Sveučilište Jurja Dobrile u Puli
Fakultet ekonomije i turizma
«Dr. Mijo Mirković»

SARA NIKOLAŠEVIĆ

METODE I PROGRAMI ZA RUDARENJE PODATAKA

Završni rad

Pula, 2016.

Sveučilište Jurja Dobrile u Puli
Fakultet ekonomije i turizma
«Dr. Mijo Mirković»

SARA NIKOLAŠEVIĆ

METODE I PROGRAMI ZA RUDARENJE PODATAKA

Završni rad

JMBAG: 0303038038, redovita studentica

Studijski smjer: Informatika

Predmet: Sustavi temeljeni na znanju

Mentorica: Prof. dr. sc. Vanja Bevanda

Pula, prosinac 2016.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani _____, kandidat za prvostupnika informatike, smjera _____ ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____

U Puli, _____, _____ godine

IZJAVA

o korištenju autorskog djela

Ja, _____ dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom _____ _____ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____ (datum)

Potpis

_____

Sadržaj

1. UVOD

Cilj ovog rada jest objasniti određene metode za rudarenje podataka, te usporedba odabrane metode kroz tri odabrana alata za rudarenje podataka. S obzirom na količinu podataka kojima određena poduzeća u današnje vrijeme raspolažu, termin rudarenja podataka ima veliku važnost. Termin rudarenja podataka možemo definirati kao pronalaženje zakonitosti i veza, te otkrivanje znanja i važnih informacija iz velikih količina podataka. Znanje i zakonitosti iz podataka pronalazimo primjenom različitih metoda. Izbor metode ovisi o području primjene, odnosno o problemu koji se rješava. Primjenom rudarenja podataka brže donosimo poslovne odluke s obzirom na rezultate analize. Metode opisane u ovom radu su metoda potrošačke košarice, memorijski temeljno razlučivanje, klasteriranje, stabla odlučivanja te naivni Bayesov klasifikator.

Metoda potrošačke košarice se koristi za otkrivanje asocijativnih pravila kako bi se vidjelo koji se parovi artikala pojavljuju na maloprodajnom računu zajedno, odnosno kupuju zajedno. U pod poglavlju ove metode opisan je a priori algoritam koji se najčešće koristi u procesu analize potrošačke košarice, te stablo frekventivnih uzoraka.

Memorijski temeljno razlučivanje je metoda koja se koristi za pronalaženje sličnosti među kategorijama. Ona je građevni element metode klasteriranja.

Klasteriranje je metoda čiji algoritmi za klasteriranje traže sličnosti unutar zadane populacije te ih grupiraju na osnovu zajedničkih karakteristika u klastere. U pod poglavlju klasteriranja opisano je K – means klasteriranje, te hijerarhijsko klasteriranje.

Stabla odlučivanja koriste se za razvrstavanje, predviđanje, grupiranje, opisivanje podataka i vizualizaciju. Stablo odlučivanja je izgrađeno od čvorova, grana te listova. U pod poglavlju ove metode ukratko je opisan rad ID3 algoritma te C4.5 algoritma.

Bayesov klasifikator je algoritam koji pretpostavlja da su svi atributi neovisni jedan od drugog te jednako važni. Počiva na teoremu uvjetne vjerojatnosti.

Kod opisa tri izdvojena alata, RapidMiner, Weka te Orange napravljena je analiza korištenjem metoda rudarenja podataka te objavljeni rezultati procesa.

2. RUDARENJE PODATAKA

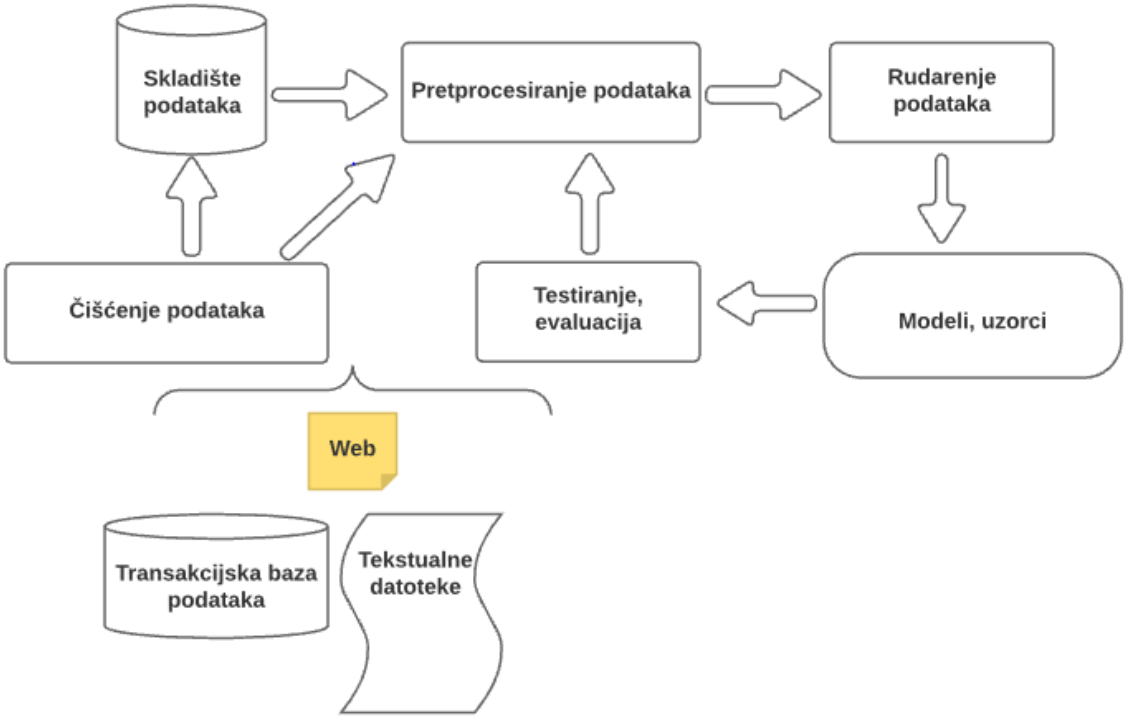
Rudarenje podatka (eng. Data mining) je pronalaženje zakonitosti u podacima. Podaci mogu biti organizirani u baze podataka, mogu biti tekstualni podaci, nestrukturirani podaci proizašli iz Web-a ili podaci koji su organizirani u vremenske serije. Primjenom raznih metoda za rudarenje podataka pronalazimo zakonitosti u podacima. Statistika, matematika, teorija informacija te umjetna inteligencija samo su neka od područja iz kojih potječu korijeni tih metoda.[1]

Tradicionalno rudarenje podataka podrazumijeva tradicionalne baze kao izvore podataka, a metode se primjenjuju nad tako formatiranim podacima. U novije vrijeme izdvajaju se pod područja s obzirom na izvore podataka kao što je to rudarenje Weba, rudarenje teksta, te analiza vremenskih serija. Osnovni razlog izdvajanja ovih područja proizlazi iz činjenice što podaci nisu strukturirani u relacijske tablice, već su nestrukturirani, ili pak strukturirani na temelju specifičnog formata. Čišćenje podataka jest jedan od osnovnih procesa u metodici otkrivanja znanja, bez obzira na izvor podataka. U tom se procesu podaci filtriraju od nečistoća, a posebna se pozornost obraća na analizu mogućih ekstremnih, tzv. stršećih vrijednosti koje ne moraju nužno biti šum u podacima već vrlo vrijedan podatak.[2]

U sustavima poslovne inteligencije rudarenje podataka je podržano skladištima podataka. Ukoliko dođe do nepostojanja skladišta podataka, što je u praksi čest slučaj, tada se odmah nakon procesa čišćenja podataka ulazi u proces pred obrade podataka. Korištenje skladišta podataka prilikom rudarenja podataka ima prednost što skladišta podataka u vrlo kratkom vremenskom periodu mogu selektirati određene skupove podataka. Iako ovakav pristup štedi vrijeme analitičaru, u praksi se često ipak prakticira izravan pristup podacima.[3]

[1] Ž. Panian, G. Klepac, Poslovna inteligencija, Zagreb, Masmedia, 2003, str.247
[2] http://www.goranklepac.com
[3] Loc. Cit.

Rudarenje podataka se može primjenjivati u svim područjima u kojima raspolažemo sa velikom količinom podataka iz domene tog područja i na osnovu tih podataka želimo otkriti određene pravilnosti, veze i zakonitosti. Ta područja mogu biti ekonomija, medicina, genetika, mehanika i druga područja. S obzirom na to da postoji čitav niz faktora koji mogu utjecati na neki određeni događaj, odnosno ishod istog, zadatak rudarenja podataka jest otkriti najznačajnije faktore i njihova obilježja u odnosu na ciljano stanje.[4]



Slika 1. Otkrivanje znanja primjenom metoda rudarenja podataka 5

Rudarenje podataka je traženje novih i vrijednih informacija u velikim količinama podataka. Omogućeno je kroz suradnju čovjeka i kompjutora. Najbolji rezultati se dobivaju upravo balansom između znanja stručnjaka koji opisuje problem i mogućnostima traženja kompjutora. U praksi, primarni ciljevi rudarenja podataka jesu

[4] Loc. cit
[5] Panian Željko, Klepec Goran, str.249

predviđanje i opisivanje. Predviđanje uključuje korištenje određenih varijabli ili polja u skupovima podataka kako bi se predvidjeli nepoznate ili buduće vrijednosti drugih potrebnih varijabli. Opisivanje se odnosi na pronalaženje obrazaca na način da se opisuju podaci koje čovjek može interpretirati. Aktivnosti rudarenja podataka se mogu smjestiti u jednu od dvije kategorije[6]:

1. Prediktivno rudarenje podataka, koje proizvodi model sistema na osnovu raspoloživih skupova podataka, ili
2. Deskriptivno rudarenje podataka, koje proizvodi nove, netrivijalne informacije temeljene na dostupnim skupovima podataka.

Krajnji cilj prediktivnog rudarenja podataka jest proizvodnja modela, koji je izražen kao izvršni kod, koji se može koristiti za izvođenje klasifikacije, predikcije, procjene ili drugih sličnih zadataka.

Krajnji cilj deskriptivnog rudarenja podataka jest da se dobije razumijevanje analiziranog sistema otkrivanjem obrazaca i veza unutar velikih skupova podataka. Ciljevi prediktivnog i deskriptivnog rudarenja podataka postignuti su korištenjem tehnika/metoda rudarenja podataka, koje su objašnjenje kasnije u radu, za sljedeće primarne zadatke rudarenja podataka[7]:

1. Klasifikacija: otkrivanje funkcija prediktivnog učenja koje klasificira podatak u jednu od predefiniranih klasa.
2. Regresija: otkrivanje funkcija prediktivnog učenja koje smještaju podatak u prediktivnu varijablu koja ima stvarnu vrijednost.
3. Klasteriranje: deskriptivni zadatak u kojem jedan nastoji identificirati završni skup kategorija ili klastera kako bi se opisao podatak.
4. Sumarizacija: dodatni deskriptivni zadatak koji uključuje metode za pronalaženje kompaktnih opisa za skupove podataka.

[6] M. Kantardžić, Data mining concepts, models, methods and algorothms, Kanada, John Wiley & SOns, Inc.,Hoboken,New Jersey, 2011, str.2
[7] Ibidem, str.3

5. Ovisno modeliranje (eng. Dependency modeling): traženje lokalnog modela koji opisuje značajne ovisnosti između varijabli ili između vrijednosti neke značajke unutar skupa podataka ili dijelovima skupa podataka.
6. Promjena i otkrivanje devijacija: otkrivanje najznačajnijih promjena unutar skupova podataka.

2.1.Korijeni rudarenja podataka

Većina problema rudarenja podataka i odgovarajućih solucija imaju korijenje u klasičnoj analizi podataka. Rudarenje podataka se primjenjivalo u mnogim disciplinama, od kojih su dva najvažnija, a to su statistika i strojno učenje(eng. machine learning). Statistika ima svoje korijenje u matematici, iz tog razloga je bio naglasak na matematičkoj strogosti, želja za dokazivanjem da je nešto razumno na teorijskim osnovama prije testiranja u praksi. Strojno učenje sadrži svoje osnove u praksi računala. Moderna statistika se gotovo u potpunosti izvodi na ideji modela. Struktura je osnovana na hipotezi , koja vodi do podatka. Kako statistika ima naglasak na modelima, tako strojno učenje daje naglasak algoritmima. Osnovni principi modeliranja u rudarenju podacima također imaju korijene u teoriji kontrole (eng. control theory), koja je primarno primijenjena u inženjerskim sistemima i industrijskim procesima. Problem određivanja matematičkog modela za neki nepoznati sustav koji opaža njegove ulazne i izlazne parove podataka se generalno odnosi na identifikaciju sustava.[8]

Identificiranje sustava koristi se u mnogo svrha, a sa stajališta rudarenja podataka, najvažnije je u svrhu predviđanja ponašanja sustava i objašnjenja interakcija i veza između varijabli koje pripadaju sustavu.

Identifikacija sustava generalno uključuje dva top-down koraka:[9]

1. Identifikacija strukture. U ovom koraku, primjenjujemo a priori znanje o cilju sustava kako bi se kako bi se odredile klase modela unutar kojih će potraga za

[8] Ibidem, str.4
[9] Ibidem, str.5

najpodobnijim modelom biti provedena. Često je ta klasa modela obilježena parametriziranom funkcijom y = f(u,t), gdje je y izlaz modela, u je ulazni vektor, i t je vektor parametra. Determinacija funkcije f je problem-ovisna, i funkcija je bazirana/osnovana na iskustvu dizajnera, intuiciji, i zakonu prirode upravlja ciljem sustava.

2. Identifikacija parametra. U drugom koraku, kad je struktura modela poznata, sve što trebamo jest primijeniti tehnike optimizacije kako bi determinirali vektor parametar t tako da rezultat modela y* = f(u,t*) može opisati sustav na odgovarajući način.

Identifikacija sustava nije proces koji se odvija u jednom smjeru. I struktura i parametar identifikacije se moraju ponavljati dok nije nađen zadovoljavajući model. Tipični koraci u svakoj iteraciji su sljedeći:[10]

1. Odrediti i parametarizirati klasu matematičkih modela, y* = f(u, t*), koji predstavljaju sustav koji se identificira.
2. Provesti identifikaciju parametara kako bi izabrali parametre koji najbolje pristaju dostupnom skupu podataka (razlika y – y* je minimalna).
3. Obaviti testove validacije kako bi vidjeli da li model identificira odgovore točno na nevidljivim skupovima podataka.
4. Završiti proces kada je rezultat validacije testa zadovoljavajući.

2.2.Veliki skupovi podataka

Količina podataka sa kojima raspolažemo je često prevelika da bi se obradili ručnom analizom, čak i za neke analize koje su bazirane na korištenju računala. Logično je da će neka osoba, odnosno voditelj raditi efektivnije ukoliko raspolaže sa velikom količinom podataka, nekoliko stotina ili tisuća podataka u arhivi. Poslovno društvo je svjesno sa današnjim problemom velike opterećenosti informacijama, te jedna analiza pokazuje sljedeće:[11]

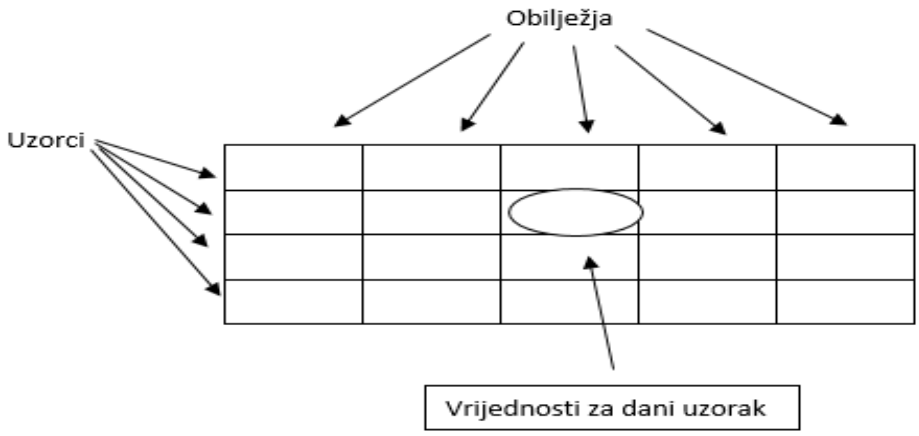
[10] Ibidem, str.5
[11] Ibidem, str.11

1. 61% menadžera vjeruje kako je opterećenje informacija prisutno u njihovom radnom okruženju
2. 80% vjeruje kako će situacija postati još lošija
3. Preko 50% menadžera ignoriraju podatke u trenutnom procesu donošenja odluka zbog preopterećenja informacija
4. 84% menadžera pohranjuje te podatke za budućnost, ne koriste ih u aktualnim analizama, i
5. 60% vjeruje da je trošak prikupljanja informacija veći od same vrijednosti.

Najprihvatljivije rješenje na probleme ove analize jest zamjena klasičnih analiza i interpretacije podataka sa novim tehnologijama rudarenja podataka. U teoriji, kod većine metoda koje se primjenjuju za rudarenje podacima trebali bi biti zadovoljni sa velikim skupom podataka, iz tog razloga što kad raspolažemo većim skupom podataka tada postoji veća mogućnost za prikupljanje vrijednih informacija, kao što i sama definicija kaže da je rudarenje podataka traženje vrijednih informacija u velikim količinama podataka.

Različiti tipovi podataka su generirani i digitalno pohranjeni u današnjem multimedijskom okruženju koji ima veliku infrastrukturu interneta. Kako bi se odabrale odgovarajuće metode za rudarenje podataka, moraju se analizirati osnovni tipovi i karakteristike skupova podataka. Prvi korak u toj analizi je sistematizacija podatka u odnosu na kompjuterski prikaz i uporabu. Podaci koji su obično izvor za proces rudarenja podataka mogu biti klasificirani u strukturirane, polu strukturirane i nestrukturirane podatke.

Većina poslovnih baza podataka sadrže strukturirane podatke koji se sastoje od dobro definiranih polja sa numeričkim ili alfanumeričkim vrijednostima, dok znanstvene baze podataka mogu sadržavati sve tri klase podataka.



Slika 2. Tablični prikaz skupa podataka12

Primjeri polu strukturiranih podataka su elektroničke slike poslovnih dokumenata, medicinska izvješća, izvršni sažeci i slično. Većina Web dokumenata također spada u ovu kategoriju. Primjeri nestrukturiranih podataka je video snimljen nadzornom kamerom u nekom odjelu. Ova forma podatka generalno zahtjeva opsežnu analizu kako bi se izdvojila i strukturirala informacija koja je sadržana u njemu. Strukturirani podaci se obično odnose na tradicionalne podatke, dok polu strukturirani i nestrukturirani podaci zajedno spadaju u netradicionalne podatke (multimedijski podaci). Većina trenutnih metoda rudarenja podataka i komercijalnih alata se primjenjuje na tradicionalne podatke.[13]

U literaturi rudarenja podataka koristi se pojam uzorak(eng. sample) ili slučaj(eng. case) za redove. Mnogo različitih tipova obilježja (atributi ili varijable), što su, polja u arhivi strukturiranih podataka su zajednička u rudarenju podataka. Nisu sve metode za rudarenje podataka jednako dobro kad je riječ o različitim tipovima obilježja.

Današnja računala te odgovarajući softverski alati podržavaju procesiranje skupova podataka koji imaju milijune uzoraka i stotine obilježja. Veliki skupovi podataka, uključujući one sa miješanim tipovima podataka, su tipično inicijalno okruženje aplikacija za tehnike rudarenja podataka. Kada je velika količina podataka

[12] Ibidem, str.12
[13] Ibidem, str. 12

pohranjena u računalu ne mogu se odmah početi primjenjivati tehnike rudarenja podataka, iz razloga što se prvo mora riješiti važan problem kvalitete podataka. Također, očito je da kvalitetna ručna analiza nije moguća u toj fazi. Iz tog razloga je vrlo važno pripremiti analizu kvalitete podataka u ranijim fazama procesa rudarenja podataka, obično je to zadatak koji se rješava u fazi pred procesiranja podataka. Kvaliteta podataka može ograničavati mogućnosti krajnjih korisnika za donošenje informiranih odluka. Postoji nekoliko indikatora kvalitete podataka koji moraju biti riješeni u fazi pred procesiranja u procesu rudarenja podataka, a to su, podaci moraju biti točni, pohranjeni, podaci trebaju imati integritet, moraju biti dosljedni, ne smiju biti redundantni, moraju biti pravovremeni, moraju biti razumljivi te moraju biti kompletni.[14]

[14] Ibidem, str.13

3. ČIŠĆENJE, PRETPROCESIRANJE I SKLADIŠTA PODATAKA ZA RUDARENJE PODACIMA

Izvori podataka imaju važnu ulogu kada govorimo o kvaliteti podataka, a o uspješnosti rudarenja podataka ovisi upravo kvaliteta podataka. No, osim izvora podataka, vrlo važnu ulogu u kvaliteti podataka imaju postupak čišćenja podataka te pretprocesiranja. Korištenje skladišta podataka kao izvor podataka za analizu predstavlja veliku prednost sustava poslovne inteligencije iz razloga što štedi vrijeme analitičaru. Kada govorimo o pretprocesiranju podataka najznačajniji metodološki postupci su pronalaženje ekstremnih vrijednosti, dijagnosticiranje vrijednosti koje nedostaju te njihovo predviđanje, povezivanje relacijskih ključeva iz različitih izvora podataka, postizanje konzistentnosti podataka, uzorkovanje, kategoriziranje vrijednosti atributa, formiranje izvedenih atributa (eng. Binning), grupiranje odnosno sažimanje podataka te normiranje.[15]

Primarni cilj skladišta podataka je da se poveća „inteligencija“ procesa odlučivanja i znanje ljudi koji su uključeni u taj proces. Neke definicije skladišta podataka su ograničene na podatke, a neke se odnose na ljude, procese, softvere, alate i podatke. Jedna od globalnih definicija glasi[16]:

„*Skladište podataka je kolekcija integriranih, subjektno orijentiranih baza podataka, koje su dizajnirane kako bi podržale funkcije potpore odlučivanju, gdje je svaka jedinica podatka relevantna nekom trenutku vremena*.“

Gledajući na ovu definiciju, na skladište podataka može se gledati kao repozitorij sa podacima organizacije, utvrđen kako bi podržao strateško donošenje odluka. Funkcija skladišta podataka jest da pohranjuje povijesne podatke organizacije na integriran način koji odražava razne aspekte organizacije i poslovanja. Podaci u skladištu podataka nisu nikad ažurirani već se koriste samo za odgovaranje upitima

[15] Panian Željko, Klepec Goran, str.252
[16] Ibidem, str. 14

krajnjih korisnika koji su generalno donositelji odluka. Dva aspekta skladišta podataka koja su najvažnija za bolje razumijevanje njegovog procesa dizajna su:

1. Specifični tipovi (klasifikacija) podataka pohranjeni u skladište podataka
2. Skup transformacija korištenih kako bi se pripremili podaci u finalnu formu koju je moguće koristiti kod donošenja odluka.

Skladište podataka uključuje sljedeće kategorije podataka, gdje je klasifikacija prilagođena izvorima podataka koji su vremenski ovisni[17]:

1. Stari podaci(eng. old detail data)
2. Trenutni (novi) podaci
3. Površno sažeti podaci
4. Visoko sažeti podaci
5. Meta-data

Kako bi se pripremili ovih pet tipova izvedenih podataka iz skladišta podataka, standardizirani su osnovni tipovi transformacije podataka. Postoji četiri osnovnih tipova transformacije, i svaki ima svoje karakteristike[18]:

1. Jednostavna transformacija. Ovakva transformacija podrazumijeva izgrađene blokove svih drugih transformacija, koje su kompleksnije. Ova kategorija uključuje manipulaciju podatkom koji je fokusiran na jedno polje u vremenu, bez pristupanja vrijednostima iz povezanih polja.
2. Čišćenje. Ovakva transformacija osiguravaju dosljedno formatiranje (nepromjenjivo) te korištenje nekog polja ili povezanih grupa polja. Ovo može uključivati valjano formatiranje adresa. Ova kategorija također uključuje provjere valjanih vrijednosti u određenom polju/području.

[17] Ibidem, str.15
[18] Ibidem, str.15

3. Integracija. Ovo je proces uzimanja operativnih podataka iz jednog ili više izvora te njihovo raspoređivanje, jedno po jedno polje, u novu strukturu podataka u skladištu podataka.
4. Agregacija i sažimanje. Ovo su metode kondenziranja instanci podataka pronađenih u operativnom okruženju u manje instanci u okruženju skladišta.

Sažimanje je dodavanje vrijednosti kroz jednu ili više dimenzija podatka, na primjer, dodavanje dnevne prodaje kako bi se dobila mjesečna prodaja. Agregacija se odnosi na dodavanje različitih poslovnih elemenata u zajednički iznos, na primjer, dodavanje dnevne prodaje proizvoda i mjesečne prodaje kako bi se dobila kombinirana, totalna mjesečna prodaja.

Ove transformacije su glavni razlog zašto preferiramo skladište kao izvor podataka za proces rudarenja podacima. Ukoliko je skladište podataka dostupno, faza pred procesiranja rudarenja podacima je značajno smanjena, ponekad čak i eliminirana. Proces razvijanja skladišta podataka je sažeto kroz tri koraka[19].

1. Modeliranje. Odvaja se vrijeme kako bi se razumjeli poslovni procesi, informacijski zahtjevi tih procesa te odluke koje su donesene unutar tih procesa.
2. Izgradnja.
3. Razmještanje. Implementacija podatka koji će biti skladišten i raznih alata poslovne inteligencije.

Rudarenje podataka predstavlja jednu od osnovnih aplikacija za skladištenje podataka, s obzirom na to da je sama funkcija skladišta podataka pružanje informacija krajnjim korisnicima.

[19] Ibidem, str.16

4. METODE RUDARENJA PODATAKA

Prve metode u svojem osnovnom obliku, koje su deklarirane kao metode rudarenja podataka razvijene su 1970 – ih i 1980-ih godina. Definicija rudarenja podataka (eng. Data mining), koja je predstavljala skup metoda koje za glavni cilj imaju otkrivanje znanja, odnosno zakonitosti u velikoj količini podataka, nastala je tek polovicom 1990 – ih godina. Proučavanje ovog područja se oslanja na više različitih znanstvenih disciplina, te je iz tog razloga teško predstaviti određene metode kao isključive metode za rudarenje podataka.

Bez obzira na interdisciplinarnost područja, postoji opće prihvaćen skup metoda koje su deklarirane kao metode rudarenja podataka, no postoji i mnogo metoda koje se mogu svrstati i u neko drugo srodno područje. Analitičar koji se bavi istraživanjem podataka, odnosno traženjem zakonitosti u podacima metodama rudarenja podataka mora imati mnogo znanja o svakoj metodi, odnosno mora poznavati pojedinosti svake metode kako bi kvalitetnije obradio podatke sa potpunim razumijevanjem. To je nužno iz razloga što je osnovni korak priprema podataka, a kako bi to bilo moguće mora poznavati sljedeće procese u analizi.

Postoji mnogo metoda rudarenja podataka. U ovom radu objasniti ću najkorištenije odnosno najznačajnije metode a to su metoda potrošačke košarice, memorijski temeljno razlučivanje, klasteriranje stabla odlučivanja, Bayesove mreže, neutronske mreže, neizrazita logika, genetički algoritmi i genetičko programiranje.

Prilikom svake analize rudarenja podataka pretpostavlja se dijeljenje populacije na uzorak za učenje i uzorak za testiranje. Na uzorku za učenje algoritmi pokušavaju na temelju podataka raspoznati uzorke, pravilnosti, vrijednosti koeficijenta postavljenog modela. Uzorak za testiranje koristi se kako bi se nakon treninga provjerila pouzdanost dobivenog rješenja. Nakon provedene analize koja se sastoji od učenja na uzorku za učenje i testiranja na uzorku za testiranje, analitičar dobiva informacije o pouzdanosti

modela. Procjeni li se da je model (ili niz modela) nepouzdan, moguće je mijenjati parametre izabrane metode ili niza metoda, ali i same metode[20].

Prema Željku Panianu i Goranu Klepacu rudarenje podataka u okviru sustava poslovne inteligencije može se promatrati iz dva osnovna kuta:

1. Prvog, koji podrazumijeva da u okviru sustava poslovne inteligencije djeluje analističar koji metodama rudarenja podataka analizira podatke, formira modele, kreira izvještaje na temelju analiza koji sadrže rezultate istraživanja zanimljive zainteresiranim stranama, interpretira dobivene rezultate analize, te sugerira daljnje akcije na temelju dobivenih rezultata analize.
2. Drugog, koji pretpostavlja da u okviru sustava poslovne inteligencije postoje unaprijed formirani moduli za analizu temeljeni na metodama rudarenja podataka, a koriste ih menadžeri kao potporu odlučivanju.

4.1. Metoda potrošačke košarice

Metoda potrošačke košarice predstavlja otkrivanje asocijativnih pravila koja prikazuju koji se parovi artikala i s kojim vjerojatnošću kupuju zajedno, što se matematički opisuje kao asocijativno pravilo $x^{\rightarrow}$y, gdje je $\mathrm{x} \cap \mathrm{y} = 0$. Mjera podrške (eng. Support) skupa artikala X je omjer transakcija koje sadrže skup transakcija (primjerice, X i Y) u odnosu na ukupan broj transakcija. Pouzdanost (eng. Confidence) se može definirati kao postotak transakcija koje, ako sadrže artikl X onda sadrže i artikl Y, u odnosu na sve transakcije koje sadrže X[21]:

$$\text{Pouzdanost } (X \rightarrow Y) = \text{podrška } (\mathrm{x} \cup \mathrm{y}) \,/\, \text{podrška}(X)$$

Koristeći uvjetnu vjerojatnost, pouzdanost se može izraziti kao:

$$\text{Pouzdanost}(X \rightarrow Y) = \mathrm{P}(\mathrm{Y}|\mathrm{X}) = P(X \cap \mathrm{Y}) \,/\, \mathrm{P}(\mathrm{X})$$

Analiza potrošačke košarice otkriva skrivena pravila u nizu takvih transakcija koja se tiču prodaje robe. U literaturi se ova metoda često naziva kolaborativno filtriranje.

[20] Panian Ž., G. Klepac, Poslovna inteligencija, str. 277
[21] Ibidem, str. 280

Cilj analize potrošačke košarice je otkrivanje pravila koja prikazuju vjerojatnost da će kupac kupiti proizvod Y ako je kupuje proizvod X. Ta vjerojatnost je temeljena na povijesnim podacima transakcijske baze. U bazi podataka postoje informacije o izvršenim transakcijama, a svaki račun u bazi predstavlja jednog kupca. Maloprodajni račun također prikazuje skup proizvoda koje je taj kupac kupio u odgovarajućem maloprodajnom centru. To je ilustrirano sljedećom tablicom[22]:

Kupac	Artikli
1	Mlijeko, kava
2	Pahuljice, mlijeko, šećer
3	Mlijeko
4	Mlijeko, kava
5	Šećer, kava

Tablica 1. Tablični prikaz skupa proizvoda maloprodajnog računa kupca

Prvi korak kod korištenja ove metode je određivanje matrice pojavnosti, što bi za prethodni primjer bilo:

	Mlijeko	Šećer	Pahuljice	Kava
Mlijeko	4	1	1	1
Šećer	1	2	1	0
Pahuljice	1	1	1	0
Kava	2	1	0	1

Tablica 2. Matrica pojavnosti skupa proizvoda maloprodajnog računa

Na osnovu ove tablice možemo vidjeti da u X% slučajeva kupac koji kupuje šećer kupuje i mlijeko, kupac koji kupuje kavu ne kupuje pahuljice.

Kod primjene ove metode, postoji određeni stupanj slobode pri utvrđivanju pravila s obzirom na vjerojatnosti koje se pojavljuju u transakcijama. Te transakcije su povezane s pojavama parova proizvoda. Generiranje pravila u oblicima jest temeljna ideja ove metode:

IF a THEN b

[22] Ibidem, str. 281

IF c AND d THEN e

Varijable a, b, c, d i e predstavljaju prividne varijable, proizvode. U ovom primjeru imamo dvije tvrdnje. I prva i druga tvrdnja se pojavljuje sa određenim stupnjem pouzdanosti, koji predstavlja omjer između broja pojavljivanja konkretne tražene varijable u ukupnom broju transakcija. U slučaju kad imamo pravilo IF a THEN b, stupanj pouzdanosti u transakciji predstavlja vjerojatnosti pojavljivanja varijable b ako sudjeluje varijabla a. pravilo koje ima najveći stupanj vjerojatnosti je najpouzdanije. Uzmimo za primjer pravilo IF a AND b THEN c sa stupnjem pouzdanosti 0.8. U tom slučaju pravilo IF a AND b THEN NOT c ima stupanj pouzdanosti 0.2. Nakon što podatke uobličimo u format koji je pogodan za obradu izrađuju se matrice vjerojatnosti koje reprezentiraju proračun vjerojatnosti. Dimenzije tih matrica ovise o broju promatranih korelacijskih elemenata, a mogu biti dvodimenzionalne, trodimenzionalne i višedimenzionalne[23].

Metoda potrošačke košarice osmišljena je prvenstveno s ciljem otkrivanja zakonitosti o kupnji skupova artikala u prodajnim centrima. Takve informacije mogu biti korisne za povećanje prodaje. Tako prodajni centri mogu, na primjer, reorganizirati robu na policama u skladu s otkrivenim zakonitostima, ili davanjem popusta na proizvod Y uz uvjet kupnje proizvoda X, gdje prodavatelj pronalazi interes u povećanju koeficijenta obrtaja robe uz manju maržu.

4.1.1. A priori algoritam

A priori algoritam se najčešće koristi u procesu analize potrošačke košarice. Povećanje broja kombinacija uzrokovanih kompleksnošću te osjetljivosti na umnožavanje elemenat analize je glavni nedostatak a priori algoritma. Metoda formiranja prividnih varijabli i metoda grupiranja skupa proizvoda na temelju zajedničkih karakteristika su metode koje reduciraju broj kandidata koji ulaze u analizu[24].

[23] Ibidem, str.282
[24] Ibidem, str. 284

Artikli	Pouzdanost		Artikli	Pouzdanost		Artikli	Pouzdanost
A1	2	Pouzdanost >=5	a2	12	Generiranje kandidata	a2,a1	1
a2	12	=>	a3	14	=>	a2,a4	4
a3	14		a6	a6		a2,a5	8
a4	3					a3,a1	2
a5	1					a3,a2	3
a6	5					a3,a4	11
						...	...

	Artikli	Pouzdanost		Artikli	Pouzdanost	
Pouzdanost >=5	a2,a5	8	Generiranje kandidata	a2,a5,a1	2	
=>	a3,a4	11	=>	a2,a5,a3	6	...
	a6,a1	7		a2,a5,a4	4	
				a2,a5,a6	2	

Slika 3. Metodologija otkrivanja asocijativnih pravila primjenom a priori algoritma25

Preciznost analize je veliki nedostatak ovog pristupa. Osnovna funkcija algoritma može se opisati u dva osnovna koraka:

- Pronalaženje frekventivnih artikala ili skupova artikala
- Generiranje asocijativnih pravila na temelju frekventivnih artikala ili skupova artikala

Redukcija broja transakcija i particioniranje baze su dvije metode za povećanje efikasnosti a priori algoritma. Metoda redukcije broja transakcija teži smanjivanju broja kombinacija te polazi od pretpostavke da transakcije koje ne sadrže nijedan k frekventni skup, ne sadrže niti k + 1 frekventni skup podataka, a metoda particioniranja baze dijeli bazu u nekoliko osnovnih particija. U svakoj particiji se računa frekvencija pojavnosti skupa artikala, te se odabiru najfrekventniji skupovi koji se pojavljuju u svim participijama[26].

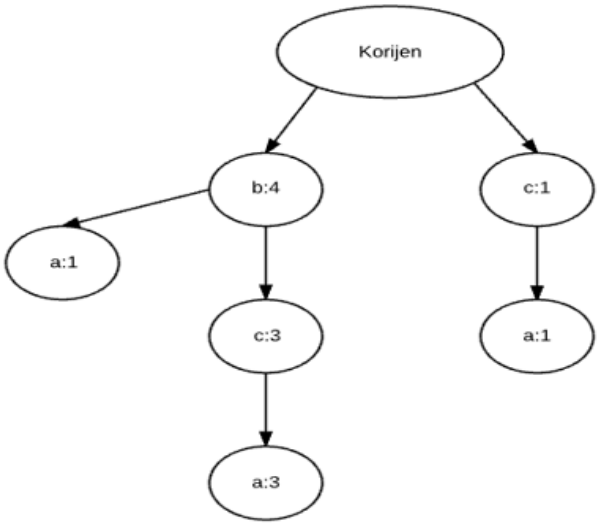
[25] Ibidem, str.284
[26] Ibidem, str.285

4.1.2. Stablo frekventivnih uzoraka

Prilikom primjene stabla frekventivnih uzoraka značajno se štedi vrijeme koje je potrebno za obradu. Stablo frekventivnih uzoraka funkcionira na način da u prolazu kroz bazu računa frekvencije pojavnosti artikala koji su sadržani u bazi te ih sortira na temelju frekvencija pojavnosti, a zanemaruje nefrekventne artikle. Na slici je prikazana metodologija otkrivanja uzoraka pomoću stabla frekventivnih uzoraka[27].

Broj transakcije	Kupljeni artikli	Artikli poredani prema frekvencijama(samo najfrekventniji artikli iz transakcija)
1	a, d, b, c	b, c, a
2	b, f, c, a, m	b, c, a
3	d, e, a, c	c, a
4	c, t, a, b, m	b, c, a
5	v, x, c, f	b, a

Uzorci
b – c – a
b – a
c - a



Slika 4. Otkrivanje uzoraka pomoću stabla frekventivnih uzoraka[28]

[27] Ibidem, str.285
[28] Ibidem, str.286

Stablo uzoraka gradi se nakon procesa sortiranja artikala prema frekvencijama pojavnosti. Stablo se gradi na način da se računaju frekvencije pojavnosti frekventivnih uzoraka te njihovih dijelova. U ovom primjeru koji je prikazan na slici, b-c-a je najfrekventniji uzorak koji se pojavljuje u 3 transakcije od mogućih 5. Uzorak c- a nije impliciran s pojavnošću b, te je izdvojen kao zaseban uzorak. Na slici možemo vidjeti i frekventni uzorak b-a koji ima isti korijen b kao i uzorak b-c-a. Iz ovog primjera možemo uočiti da ukoliko uzorak b promatramo samostalno njegova pojavnost je veća, za razliku od uzorka b-c-a, a još je veća ako ga promatramo u inačici b-a s obzirom na to da sudjeluje u obje inačice.

To možemo tumačiti na način da, pojavnost uzorka c iznosi 3, pojavnost uzorka a iznosi 3, uz uvjet pojavnosti uzorka b čija pojavnost iznosi 4. Nakon toga se formiraju pravila koja pretražuju stablo. Stablo se pretražuje na način da to znači da, ako se otkrije ista pojavnost na nižim razinama, recimo c nakon b=3 i a nakon b=3, b nakon c implicira a sa 100% - tnom vjerojatnošću. Isti princip se može primijeniti na sve grane, a grane „rezati“ od manjeg značaja prema kriteriju pojavnosti[29].

Ova je metoda izvorno zamišljena tako da se uzorci vežu sa pripadajućim frekvencijama u okviru stabla, korištenjem upitnog jezika. Korištenjem tih upita ispisivali bi se svi frekventni uzorci koji su vezani uz zadani artikl. Kvaliteta ove metode proizlazi i iz činjenice da se uz minimalno vrijeme procesiranja podataka može brzo i efikasno osvježavati već stablo koje je formirano, prilikom dobivanja novih podataka. Algoritam stabla frekventnih uzoraka može se uvelike koristiti i za prepoznavanje uzoraka u vremenskim serijama, otkrivanje uzoraka u tekstovima, te za rudarenje Weba.

Autori metode stabla frekventivnih uzoraka su Jiawei Han, Jian Pei i Yiwien Yin, a rezultat je rada na projektu kojega su podržali Vijeće prirodnih znanosti i inženjerskih istraživanja kanadske vlade, te kompanija Hewlett Packard.

[29] Ibidem, str. 287

4.2. Memorijski temeljno razlučivanje

Prema Panianu i Klepacu (2003) memorijski temeljno razlučivanje (eng. Memory Based Reasoning) je metoda koja pronalazi sličnosti među kategorijama (atributima, slogovima, unutar skupa atributa). Promatrana u širem kontekstu, ova je metoda građevni element metode klasteriranja (eng. K – mean Clustering Method). Metoda obrađuje podatke pronalaženjem vrijednosti funkcije udaljenosti između, primjerice, slogova datoteke te uspoređuje vrijednosti kombinacijske funkcije u potrazi za odgovorima o sličnosti, koje procjenjuje na temelju udaljenosti.

4.2.1.Model rada i osnovnih procesa

Pretprocesirani podaci su osnovna perspektiva promatranja ove metode. Redak u pretprocesiranoj tablici je osnovna jedinica obrade ili određeni element retka. Vrijednosti funkcije udaljenosti vrijednosti kombinacijskih funkcija se računaju za svaki redak ili element retka u toj tablici. Kod računanja udaljenosti između redaka (slogova) mogu se koristiti standardne funkcije udaljenosti, odnosno može se birati između[30]:

(a) Apsolutne vrijednosti razlike

|A - B|

(b) Normalizirane apsolutne vrijednosti (mr – maksimalna razlika)

|A - B|/(mr)

(c) Euklidske udaljenosti

$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i1} - x_{j1}|^2 + \ \dots\ |x_{iq} - x_{jq}|^2}$$

(d) Manhattan udaljenosti

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i1} - x_{j1}| + .. |x_{iq} - x_{jq}|$$

[30] Ibidem, str.289,290

Gdje su i = (xi1, xi2, … xiq), j = (xj1, xj2,..xjq) dva q – dimenzionalna podatkovna objekta.

Vrijednost normalizirane apsolutne vrijednosti se uvijek nalazi između 0 i 1. Kod računanja Euklidske udaljenosti preporučuje se normirati podatke prije obrade, metodom min – max, u intervalu <0,1>.

Ukoliko u izračunu sudjeluje više od jedne vrijednosti, tada se sve te vrijednosti udaljenosti sumiraju te se dobivena vrijednosti koristi u obradi. Kategorije predstavljaju svaki od parcijalnih elemenata, na primjer dob, vrijednost prodaje itd. Rezultat obrade je prikazan matricom, a vrijednosti prikazane u matrici su vrijednosti udaljenosti između promatranih slogova[31]:

d(1, 1)	d(1, 2)	d(1, 3)..	d(1, y)
d(2, 1)	d(2, 2)	d(2, 3)..	d(2, y)
d(3, 1)	d(3, 2)	d(3, 3)..	d(3, y)
d(x, 1)	d(x, 2)	d(x, 3)..	d(x, y)

u kojoj svaki element matrice predstavlja udaljenost između slogova. Krajnji je korak funkcija kombinacije koja može biti funkcija maksimuma ili minimuma, odnosno neka daljnja funkcija, primjerena problemu koji se rješava.

Ova metoda može biti korisna u slučajevima pronalaženja sličnosti među slogovima, odnosno kategorijama koje definiramo (kupci, proizvodi) ili pak pomaže pri predviđanju ponašanja neke kategorije (npr. novo pridošli kupac). Kod više dimenzionalnom prostoru broj dimenzija raste s brojem atributa, dok kompleksnost prikaza raste s brojem dimenzija.

[31] Ibidem, str. 290

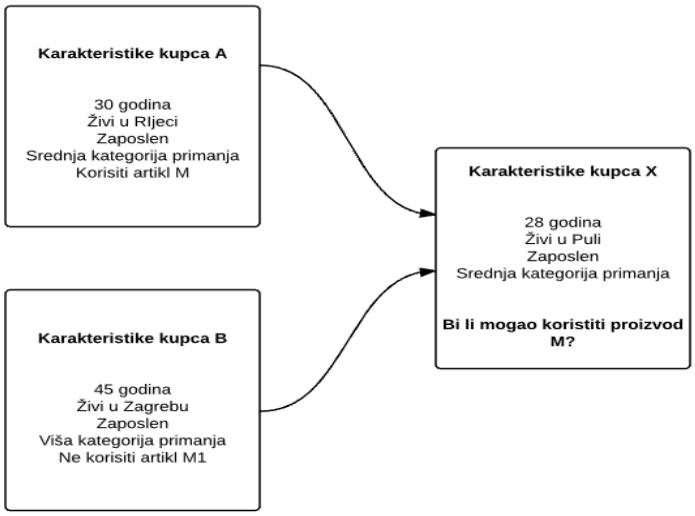
4.2.2 Funkcije udaljenosti i tipovi podataka

Osnovna ideja ove metode je transformirati ne numeričku vrijednost u numeričku vrijednost. Tako se, primjerice, kategorije tipa velik, srednji, mali mogu transformirati u vrijednosti :

Velik = 1
Srednji = 2
Mali = 3

Mogu se transformirati i segmenti tržišta, akcije koje poduzimaju kupci u određenim situacijama, marketinške kampanje i slično, pridružujući im određene vrijednosti s kojima se može ući u proces brade podataka.

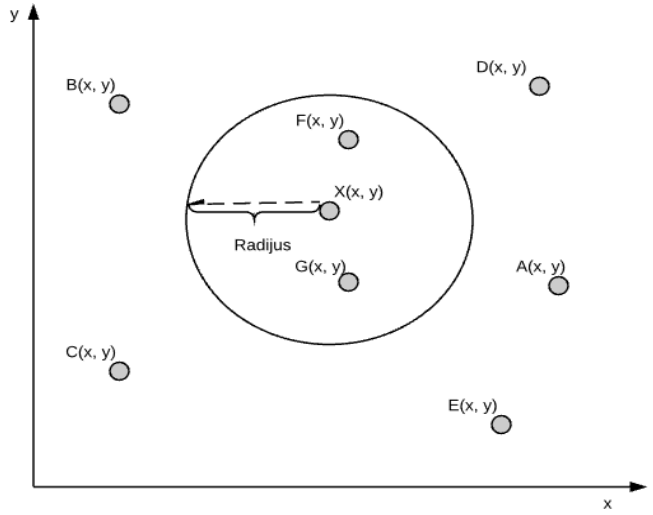
Otkrivanje sličnosti između objekata je osnovna namjena ove metode. Objekti moraju biti iskazani brojčanim vrijednostima, a te vrijednosti predstavljaju koordinate u prostoru. Segmentacija tržišta jest vrlo zahvalno područje primjene ove metode. Na temelju metode memorijski temeljnog razlučivanja postoji i mogućnost predikcije. Na slici 5 vidimo primjenu memorijski temeljnog razlučivanja u svrhu predikcije.



Slika 5. Predikcija primjenom metode memorijski utemeljenog razlučivanja[32]

[32] Ibidem, str.293

Na slici imamo kupca A, kupca B te kupca X. Procjenjuje se udaljenost kupca X prema svakom od kupaca. Na osnovu osobina kupca A i kupca B, te atributa izdvojenih za kupca X, procjenjuje se bi li kupac X mogao koristiti proizvod M. Ovo ne predstavlja problem ukoliko postoji velika sličnost unutar pojasa sličnosti, a pojas sličnosti predstavlja okolinu promatranog kupca.



Slika 6. Geometrijska interpretacija pojasa sličnosti [33]

Ako unutar pojasa sličnosti kupci ne pokazuju istu sklonost/nesklonost prema proizvodu M, sud o tome bi li kupac X mogao koristiti proizvod M može se donijeti računajući frekvencije korištenja i nekorištenja M –a u klasteru, na osnovu čega se opet može procijeniti vjerojatnost sklonosti/nesklonosti prema M-u. Vrijednost vjerojatnosti iznad koje će hipoteza biti prihvaćena ili odbačena definira analitičar. To može značiti da se na temelju vjerojatnosti <0.7 hipoteza može prihvatiti. Najnepovoljniji iznos vjerojatnosti za konkretan slučaj je 0.5, jer on ne daje nikakvu dodatnu informaciju koja bi smanjila nesigurnost donošenja odluke[34].

[33] Ibidem, str.294
[34] Ibidem, str.294

4.3. Klasteriranje

Klasteriranje se definira kao grupiranje ili objedinjavanje objekata sličnih karakteristika. Koristeći zadani skup atributa, algoritmi za klasteriranje pokušavaju pronaći sličnosti unutar zadane populacije. Koristi se Euklidska ili Manhattan udaljenost za računanje sličnosti između članova populacije nad kojom se vrši analiza[35]. Grupe se formiraju postupkom dijeljenja skupa podataka, pri čemu se pripadnost grupi definira na temelju značajki sličnih obilježja (npr., dob, spol, županija). Algoritmi za klasteriranje pokušavaju pronaći sličnosti unutar zadane populacije koristeći zadani skup atributa[36]. Postoji mnogo algoritama za klasteriranje međutim najpoznatiji je K – means algoritam, koji pomoću funkcija za procjenu distance i centroida, u iterativnom postupku kreira klastere, te aglomerativni hijerarhijski algoritam.

4.3.1. K – means klasteriranje

Ova metoda funkcionira na način da dijeli osnovnu populaciju na k segmente. Svaki od segmenata sadrži n sličnih elemenata. Na temelju funkcije udaljenosti algoritam procjenjuje sličnost elemenata. Ova metoda se algoritamski može prikazati na sljedeći način[37]:

1. Izaberi proizvoljno k segmenata (klastera).
2. Odredi središte („centroid“) za svaki od k segmenata.
3. Ponavljaj:
    - Pridruži pomoću funkcije udaljenosti sve elemente populacije njihovim najbližim klasterima (proračun se vrši na temelju centralnih vrijednosti, centroida)

[35] Ibidem, str. 295

[36] G.Klepac, L. Mršić, Poslovna inteligencija kroz poslovne slučajeve, Biblioteka Poslovna znanja, Lider press

[37] Željko Panian, Goran Klepac, op.cit., str.295

- Proračunaj novu vrijednost središta klastera (centroida) za svaki klaster pojedinačno kao prosječnu vrijednost objekata sadržanih unutar svakog klastera
- Ponavljaj sve dok se mijenjaju vrijednosti središta klastera (vrijednosti centroida)

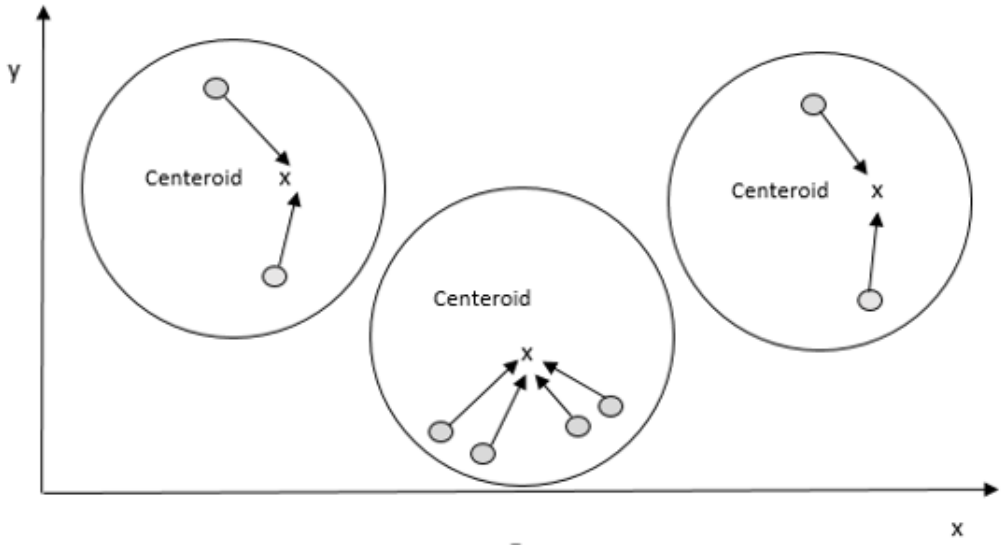
Obično se koristi kriterij kvadratne pogreške (eng. Square-error criterion) koji je definiran kao[38]:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2$$

Gdje je E suma kvadratne pogreške za sve elemente u skupu podataka, p je točka u prostoru koja predstavlja dani element, a m_i je srednja vrijednost klastera C_i (p i m_i su višedimenzionalni). To znači da za svaki element populacije u svakom klasteru, udaljenost elementa do centroida klastera se računa uz pomoć kvadratne funkcije., a udaljenosti su sumirane.

[38] J. Hann, M. Kamber, Data Mining Concepts and Techniques, university of Illinois at Urbana-Champaign, 2006, Str.402

Na sljedećoj slici je prikazan grafički prikaz K-means algoritma. Možemo vidjeti iz slike da elementi teže ka centralnim vrijednostima tzv. centroidima klastera.



Slika 7. Pojednostavljeni prikaz K-means algoritma[39]

Kod primjene ove metode analitičar izabire broj klastera, te je potrebno izvesti nekoliko iterativnih procesa klasteriranja kako bi proces broj reprezentativnih klastera bio odgovarajući, odnosno zadovoljavajući. To se smatra glavnim nedostatkom ove metode.

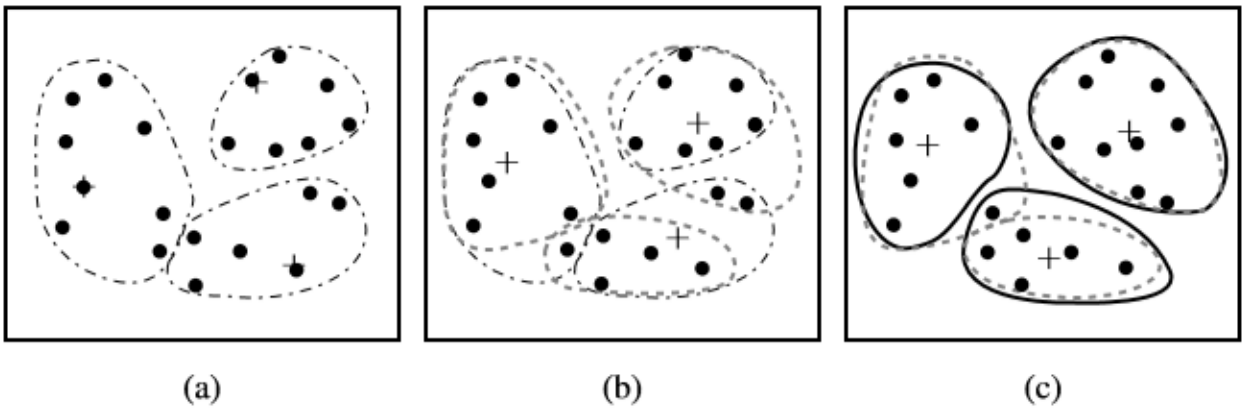
Sljedeći primjer[40] predstavlja način na koji K-means algoritam funkcionira prilikom particioniranja gdje je svaki centroid klastera predstavljen srednjom vrijednosti elemenata u klasteru. Pretpostavimo da imamo skup podataka smještenim u nekom prostoru kao što je prikazano na slici 8(a). Recimo da je broj klastera k = 3. Ulaz algoritam je k, broj klastera te D, skup podataka koji je sadržan od n elemenata. Izlaz će biti skup k klastera. Metoda glasi:

1) Proizvoljno odaberi k elemenata iz D kao početne centroide klastera
2) Ponavljaj

[39] Željko Panian, Goran Klepac, op.cit., str.296
[40] J. Hann, M. Kamber, Data Mining Concepts and Techniques, university of Illinois at Urbana-Champaign, 2006, Str.403

3) (pre)rasporedi svaki element klasteru kojem je element najsličniji, temeljeno na srednjoj vrijednosti elementa u klasteru
4) Ažuriraj srednju vrijednost klastera, to jest, izračunaj srednju vrijednost elemenata za svaki klaster
5) Sve dok nema promjene.

Na temelju tog algoritma proizvoljno se odabiru tri elementa kao tri početna centroida klastera, gdje su centroidi klastera označeni kao „+“. Svaki element je distribuiran klasteru bazirano na centroidu klastera kojem je najbliži. Takva distribucija koja je prikazana u formi kružnice prikazana je na slici 8(a).



Slika 8. klasteriranje skupa elemenata uz pomoć K - means algoritma[41].

Nakon toga centroidi klastera su ažurirani, to znači da su srednje vrijednosti svakog klastera izračunate na osnovu elemenata koji se trenutno nalaze u klasteru. Koristeći nove centroide klastera, elementi su preraspoređeni u klastere na osnovu toga koji centroid kojeg klastera im je najbliži. Takva distribucija je prikazana isprekidanom kružnicom na slici 8(b). taj se proces ponavlja te dovodi do forme koja je prikazana na slici 8(c). Na posljetku, nije potrebna redistribucija elemenata u niti jednom klasteru te se proces završava. Cilj algoritma jest otkriti k particija koji minimiziraju pogrešku kvadratne funkcije.

[41] J. Hann, M. Kamber, str. 403

4.3.2. Hijerarhijsko klasteriranje

Hijerarhijsko klasteriranje polazi od grupiranja objekata u stablo klastera. Ova se vrsta klasteriranja može klasificirati na aglomerativno i divizijsko hijerarhijsko klasteriranje. To ovisi o smjeru particioniranja, koje može biti od dna prema vrhu ili obrnuto. Nakon što se jednom izvrši podjela populacije u klastere, nemoguće je ponavljanje procesa klasifikacije na istoj razini stabla. To se smatra jednim od glavnih nedostataka hijerarhijskih algoritama za klasteriranje, a mogu se prikazati u obliku dendograma[42].

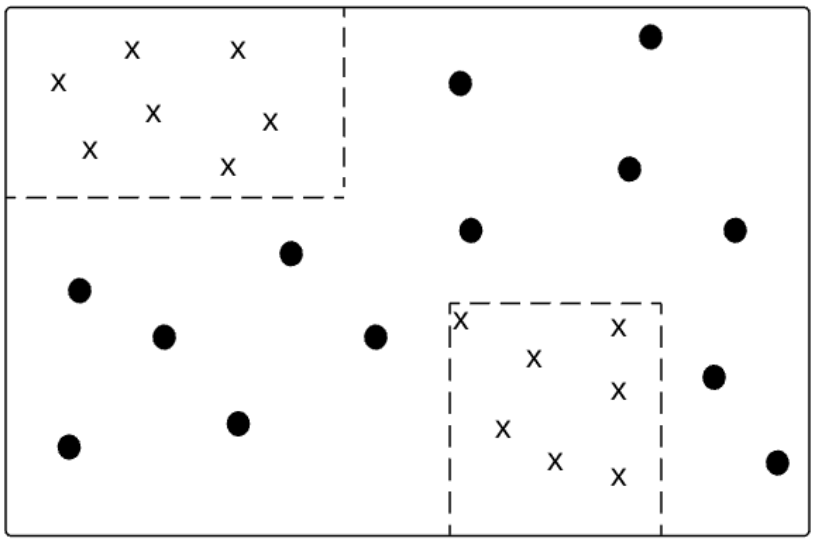
„Aglomerativno hijerarhijsko klasteriranje može se definirati kao klasteriranje metodologijom „od dna prema vrhu“ (eng. Bottom - up), svrstavanjem svakog pojedinačnog objekta u njegov vlastiti klaster. Sljedeći korak se sastoji od stvaranja novih klastera povezujući temeljne klastere u sve veće skupine, sve dok svi elementi u krajnjem koraku ne formiraju zajednički klaster, ili dok se ne ostvari uvjet prekida daljnjeg klasteriranja. Divizijsko klasteriranje se od aglomerativnog razlikuje jedino u smjeru klasteriranja koji je u ovom slučaju „od vrha prema dnu“ (eng.Top-down), pri čemu se temeljni, inicijalni jedinstveni klaster, koji sadrži sve elemente populacije, dijeli u manje klastere sve dok svaki od elemenata ne formira vlastiti klaster, ili dok se ne ispuni zadani uvjet prekida daljnjeg klasteriranja.[43] „

[42] Željko Panian, Goran Klepac, op.cit., str.298

[43] Loc.cit.

4.4. Stabla odlučivanja i pravila odlučivanja

Stabla odlučivanja i pravila odlučivanja su metode za rudarenje po podacima koje se primjenjuju za rješavanje problema klasifikacije. Klasifikacija je proces učenja funkcija koje smještaju podatke u neku od predefiniranih klasa. Svakoj klasifikaciji koja se bazira na algoritmima induktivnog učenja dodijeljen je skup uzoraka koji se sastoji od vrijednosti atributa i odgovarajuća klasa. Cilj ovakvog učenja jest kreirati klasifikacijski model, koji se naziva klasifikator. Klasifikator na osnovu vrijednosti atributa koji su mu dostupni predviđa klasu za dani uzorak. Drugim riječima, klasifikacija je proces u kojem se dodjeljuju klase nekom zapisu, a klasifikator je rezultat klasifikacije koji predviđa klasu za dani uzorak. Prilikom ovog postupka, uzorci su podijeljeni u predefinirane grupe[44].



Slika 9. klasifikacija uzoraka u 2-D prostoru[45]

Generiranje stabla odlučivanja jest vrlo učinkovita metoda stvaranja klasifikatora iz podataka te je ujedno i jedna od najkorištenijih logičkih metoda. Stablo odlučivanja

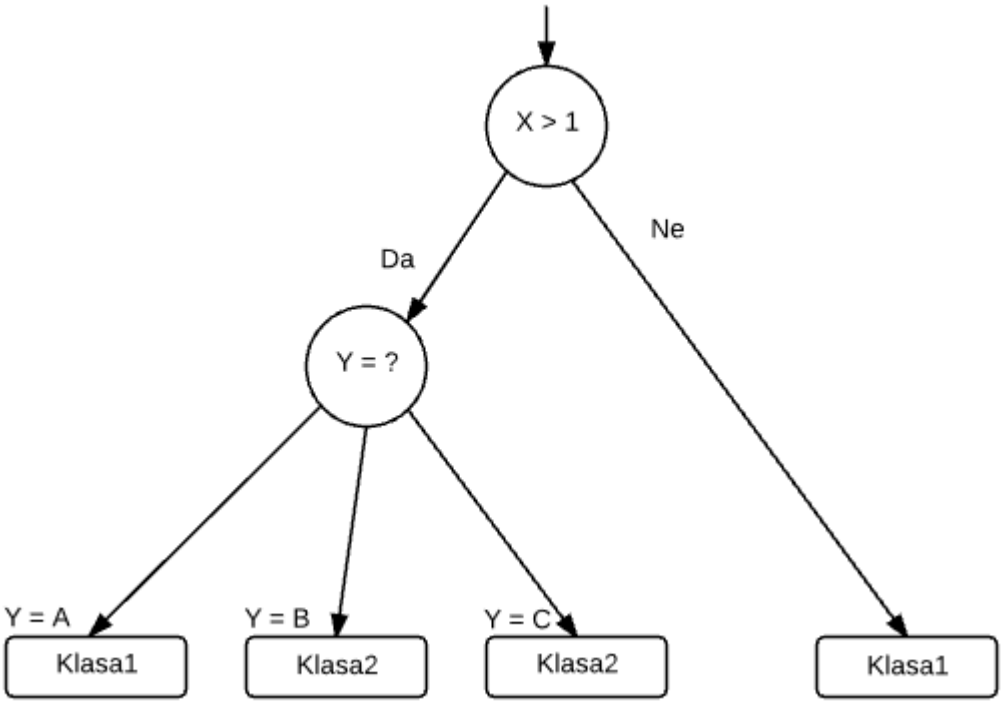
[44] Kantardžić Mehmed, op.cit., str.170
[45] Ibidem, str. 170

predstavlja hijerarhijski model, a sastoji se od čvorova i grana. Atributi se testiraju u čvorovima, a grane predstavljaju sve moguće izlaze za testirani atribut u određenom čvoru. Algoritmi stabla odlučivanja spadaju u metode nadgledanog učenja. Postoji mnogo algoritama stabla odlučivanja, kao što su ID3, C4.5, CHAID, CR&T i QUEST. Uglavnom, većina algoritama ove metode koristi tako zvanu „od vrha prema dnu“ (eng. Top-down) metodu pretraživanja. ID3 (eng. Induction of Decision Trees) je jedan od najpoznatijih algoritama koji je razvio J. Ross Quinlan te je bio temelj za proširenje C4.5 algoritma[46].

ID3 algoritam započinje sa svim uzorcima za treniranje u početnom čvoru stabla. Odabran je neki atribut kako bi podijelio te uzorke. Za svaku vrijednost atributa kreira se grana, te je odgovarajući pod skup uzoraka koji imaju vrijednost atributa specificiranu od grane, pomaknut na novo kreirani čvor dijete (eng. Child node). Algoritam se primjenjuje rekurzivno na svaki čvor dijete sve dok svi uzorci na čvoru ne pripadnu nekoj klasi. Svaki put do lista (eng. Leaf) u stablu odlučivanja predstavlja klasifikacijsko pravilo. Odabir atributa kod ID3 i C4.5 algoritma bazira se na minimiziranju mjere entropije informacije koja se primjenjuje na primjerima u čvoru. Pristup baziran na entropiji informacije predstavlja minimiziranje broja testova koji će omogućiti uzorku klasifikaciju u bazi podataka.

Jednostavan prikaz stabla odlučivanja prikazan kroz sljedeći primjer.[47] U ovom primjeru imamo dva ulazna atributa X i Y. Svi uzorci koji imaju vrijednosti $X > 1$ i $Y = B$ pripadaju „Klasa2“, dok uzorci koji imaju vrijednost $X < 1$ pripadaju „Klasa1“, koju god da vrijednost ima atribut Y.

[46] Ibidem, str. 171
[47] Ibidem, str. 172



Slika 10. Jednostavno stablo odlučivanja sa testovima nad atributima X i Y[48]

Osnovna metodologija građenja stabala sastoji se od sljedećih koraka[49]:

- Stablo započinje jedinstvenim korijenom (ciljanom varijablom) koja reprezentira cijeli uzorak.
- Ukoliko svi uzorci pripadaju istoj klasi, tada čvor postaje list i označava se tom klasom
- U protivnom koristi se mjera temeljena na entropiji (eng. Information gain) za selekciju atributa koji će najbolje razdijeliti uzorak na pod klase. Taj se atribut spominje kao testni atribut čvora. Algoritmi se u tom dijelu uglavnom razlikuju s obzirom na sposobnost operiranja različitim tipovima varijabli.
- Stablo se dalje razgranava za svaku vrijednost testnog atributa.

Koraci koji su opisani rekurzivno se ponavljaju sve dok se ne dostigne neki od kriterija koji zaustavlja rekurziju.

[48] Kantardžić Mehmed, op.cit., str.172
[49] Željko Panian, Goran Klepac, op.cit., str.304

Kako bismo precizno definirali informacijski dobitak moramo definirati entropiju. Entropija čistoća skupa primjera. Skup S sadrži pozitivne i negativne primjere nekog ciljnog koncepta, tada je entropija u odnosu na skup S:

$$Entropija(S) = -\ p_p log_2 p_p -\ p_n log_2 p_n$$

Gdje je p_p proporcija pozitivnih primjera u skupu S, a p_n proporcija negativnih primjera u skupu S. Kod svakog računanja entropije pretpostavlja se da vrijedi 0log0 = 0. ukoliko svi članovi skupa S pripadaju istoj klasi primjera tada je entropija 0. Definicija entropije koja kaže da ona specificira minimalni broj bitova informacije koji je potreban da se kodira klasifikacija bilo kojeg člana skupa S je jedna od interpretacija entropije iz teorije informacija. U slučaju kada ciljni atribut poprima više od dvije vrijednosti, u ovom primjeru a različitih vrijednosti, onda je entropija skupa S:

$$Entropija(S) = \sum_{i=1}^{a} - p_i log_2 p_i$$

Gdje je p_i proporcija klase i u skupu S. u ovom slučaju kada ciljni atribut poprima a različitih vrijednosti maksimalna entropija iznosi $log_2 a$[6].

Informacijski dobitak predstavlja očekivanu redukciju entropije koja je uzrokovana razdvajanjem primjera na osnovu tog atributa. Informacijski dobitak atributa A, u odnosu na skup primjera S:

$$Gain(S, A) = Entropija(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropija(S_v)$$

Gdje Values(A) predstavlja skup svih mogućih vrijednosti atributa A, a S_v podskup skupa S za koji atribut A ima vrijednost v. u jednadžbi za informacijsku dobit (Gain), entropija originalnog skupa S je prvi član , a drugi član je očekivana vrijednost entropije nakon što je skup S razdvojen korištenjem atributa A. Očekivana entropija opisana je drugim članom te predstavlja zbroj entropija pod skupova S_v. U donosu na formulu,

Gain (S,A) predstavlja informaciju o vrijednosti ciljnog atributa uz poznate vrijednosti atributa A[50].

4.4.1. C4.5 algoritam

Najvažniji dio ovog algoritma jest proces generiranja stabla odlučivanja iz skupa uzoraka za treniranje. Rezultat generiranja je klasifikator u formi stabla odlučivanja. Struktura se sastoji od dva tipa čvora, krajnji čvor (eng. Leaf node) koji predstavlja klasu, ili čvor odluke (eng. Decision node) koji definira uvjet u obliku vrijednosti određenog atributa iz kojeg izlaze grane koje zadovoljavaju određene uvijete. Kostur C4.5 algoritma baziran je na „Hunt's Concept Learning System“ (CLS) metodi za konstruiranje stabla odlučivanja iz skupa uzoraka za treniranje T. Recimo da kase označavamo kao $\{C_1, C_2, .., C_k\}$. Postoje tri mogućnosti za sadržaj skupa T[51]:

1. T sadrži jedan ili više uzoraka, svi pripadaju određenoj klasi C_j. Stablo odlučivanja za T je identifikacijski čvor klase C_j.
2. T ne sadrži uzorke. Stablo odlučivanja je opet čvor ali klasa koja je povezana sa čvorom mora biti utvrđena iz podataka koji nisu T, kao što bi mogla biti većina klasa u T. C4.5 algoritam koristi najčešću klasu na roditelju danog čvora.
3. T sadrži uzorke koji pripadaju različitim klasama. U ovoj situaciji ideja je da se pročisti T u pod skupove uzoraka koji ciljaju prema jednoj klasi kolekcije uzoraka. Bazirano na pojedinom atributu, odabran je odgovarajući test koji ima jedan ili više obostrano ekskluzivnih rezultata $\{O_1, O_2, \dots, O_n\}$. T je podijeljen na pod skupove $T_1, T_2, \dots, T_n$ gdje T_i sadrži sve uzorke u T koji imaju rezultat O_i odabranog testa. Stablo odlučivanja za T sastoji se od čvora odluke koji identificira test i jedne grane za svaki mogući izlaz.

[50] http://dms.irb.hr/tutorial/hr_tut_dtrees.php, pristupljeno 28.5.2016, u 12:28 h
[51] Kantardžić Mehmed, op. Cit., str.173

Originalni ID3 algoritam koristi mjeru zasnovanu na entropiji (eng. Information gain), kako bi odabrao atribute koji će biti testirani , koja je bazirana na entropiji.

4.5. Naivni Bayesov klasifikator

Ovaj algoritam pripada skupini klasifikacijskih algoritama. Pretpostavka ovog algoritma jest da atributi nisu ovisni jedan od drugog te da svi atributi imaju jednaku važnost. Primjenom metode gradijenta, treniraju se Bayesove mreže. Osnovni koncept ovog algoritma počiva na uvjetnoj vjerojatnosti. Uvjetna vjerojatnost definirana je kao[52]:

$$„P(a|b) = m“$$

Uvjetnu vjerojatnost iz ove jednadžbe možemo definirati kao vjerojatnost događaja a uz uvjet b iznosi m.

„Uvjetna vjerojatnost reducira polje slučajnih događaja, te donosi dodatnu informaciju reducirajući pri tome stupanj neizvjesnosti ishoda događaja.[53]“

Temeljno pravilo vjerojatnosti događaja x i y glasi:

$$P(x, y) = P(x|y)P(y),$$

na temelju ovog pravila proizlazi:

$$P(x|y)P(y) = P(y|x)P(x)$$

Iz čega je izvedeno Bayesovo pravilo uvjetne vjerojatnosti:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

„ Bayesove se mreže sastoje od dva osnovna elementa:

[52] Panian Ž., G. Klepac, op. Cit. Str.308
[53] Ibidem, str. 308

1. Direktnih necikličkih grafova u kojima svaki čvor predstavlja slučajnu varijablu, a svaka poveznica probabilističku zavisnost
2. Druga komponenta su tablice uvjetnih vjerojatnosti za svaku varijablu

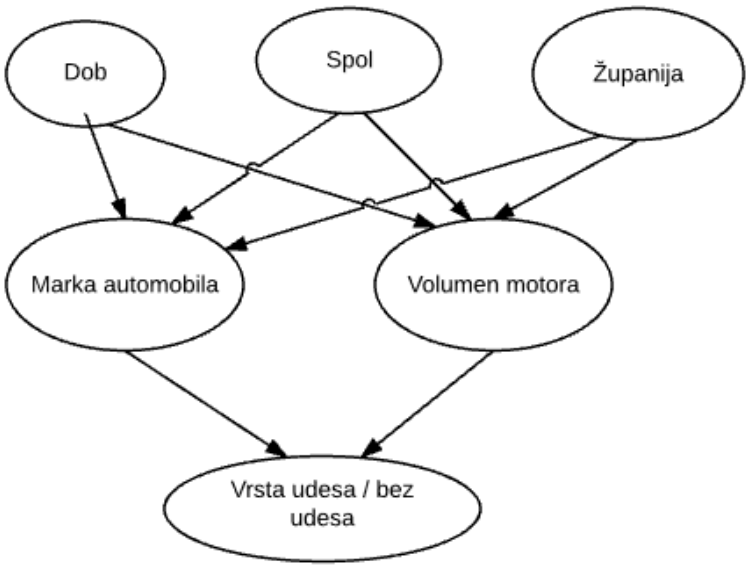
Tablica uvjetnih vjerojatnosti specificira uvjetnu distribuciju P(V|roditelj(V)).

Spojna vjerojatnost (eng. Joint Probability) atributa (V1,..Vn) koji se sastoje od vrijednosti (v1,..vn) računa se kao:

$$P(v_1, \ldots, v_n) = \prod_{i=1}^{n} P(v_i | Roditelj(V_i))$$

Gdje su elementi tablice uvjetnih vjerojatnosti $P(v_i | Roditelj(V_i))$. [54]„

Sljedeći primjer sa slikom prikazuje jednostavnu Bayesovu mrežu te tablicu uvjetnih vjerojatnosti.



Slika 11. Grafički prikaz Bayesove mreže[55]

[54] Ibidem, str. 310
[55] Ibidem, str. 133

Unutarnja struktura mreže sastoji se od tablica uvjetnih vjerojatnosti prikazano u sljedećoj tablici na primjeru varijable udesa:

Marka automobila	**Opel**	**VW**	**BMW**	**…**
Volumen motora	1.1 1.2 …	1.1 1.2 …	1.1 1.2 …	1.1 1.2 …
Bez udesa				
Lakši udes				
Srednji udes				
Teški udes				
Teški udes sa				
ljudskim žrtvama				
…				

Tablica 3. Tablica uvjetnih vjerojatnosti [56]

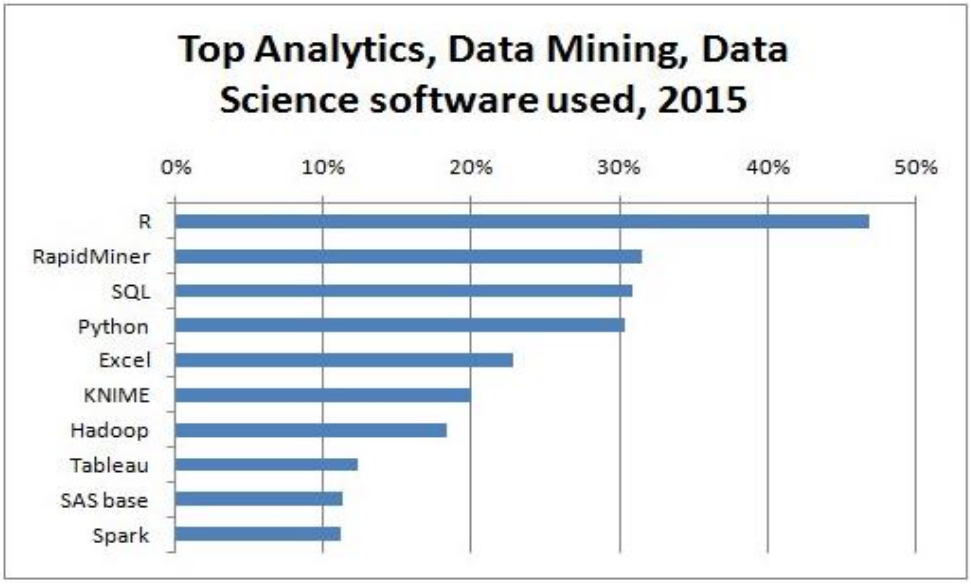
Računanjem uvjetne vjerojatnosti za svaku od prikazanih kategorija pune se elementi tablice uvjetnih vjerojatnosti vrijednostima. U ovom primjeru računa se vjerojatnost bez udesa za volumen motora 1.1 i marku automobila Opel, vjerojatnost bez udesa za volumen motora 1.2 i marku automobila Opel i tako po redu. Za svaku varijablu sa slike se konstruira tablica uvjetnih vjerojatnosti. Kategorije tipa lakši, srednji i teški udes mogu se računati na temelju brojčanih vrijednosti te je njihovo računanje dio postupka pretprocesiranja podataka[57].

Bayesove mreže se primjenjuju u sustavima poslovne inteligencije, a glavna im je karakteristika sposobnost učenja direktno iz uzoraka podataka. Uspješno se primjenjuju u ekonomiji prilikom segmentacije tržišta, za praćenje ponašanja klijenata, u medicini za dijagnosticiranje bolesti, kod razvoja softvera za traženje pogrešaka u programu, farmaciji za istraživanje lijekova, proizvodnji te poljoprivredi i stočarstvu.

[56] Ibidem, str. 310
[57] Ibidem, str. 311

5. PROGRAMI ZA RUDARENJE PODATAKA

Programi za rudarenje podacima omogućavaju nam rješavanje problema rudarenja podataka. Koristimo ih za rješavanje problema klasifikacije, klasteriranja, bayesovih mreža, asocijativnih pravila, te kod ostalih metodologija rudarenja podataka. Postoje alati koji su besplatni te komercijalni alati. Prema anketi koja je provedena 2015. godine sa stranice www.kdnuggets.com, 91% korisnika koji su sudjelovali u anketi koristi komercijalne alate, a 73% besplatne alate, oko 27% koristi samo komercijalne alate, a samo 9% koristi isključivo besplatne alate. Većina od 64% koristi i komercijalne i besplatne alate za rudarenje podacima, dok je u anketi 2014. godine, tek 49% ispitanika koristilo komercijalne i besplatne. Najpopularniji alat na cjelokupnom području rudarenja podataka i znanosti podataka jest R, a slijedi ga RapidMiner. Top deset analitičkih alata, alata za rudarenje podacima i znanost podataka korištenih u 2015. godini su R sa 46,9%, RapidMiner sa 31,5%, SQL sa 30,9%, Phyton sa 30,3%, Excel sa 22,9%, KNIME sa 20,0%, Hadoop sa 18,4%, Tableau sa 12,4%, SAS sa 11,3%, Spark sa 11,3%.



Slika 14. Top alati za analitiku, rudarenje podataka, znanost o podacima u 2015. godini, Pristupljeno 13.06.2016, 11:38, http://www.kdnuggets.com/2015/05/poll-r-rapidminer-python-big-data-spark.html

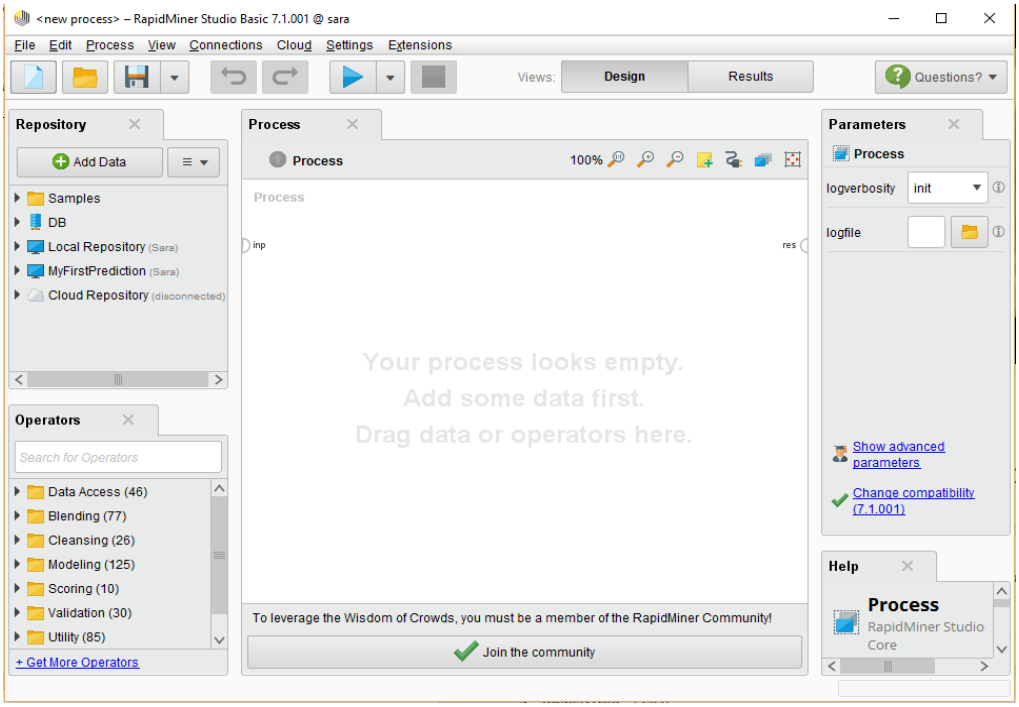
Postoje besplatni alati za rudarenje podataka, međutim postoje i oni koji su proizvedeni za prodaju te ne podržavaju besplatne verzije. Neki od takvih alata navedeni su na stranici www.kdnuggets.com, a neki od njih su AdvancedMiner, Alteryx, BayesiaLab, Civis, Data Miner SoftwareKit, SAS Enterprise Miner, Synapse i mnogi drugi.

5.1. Besplatni alati

Postoji mnogo besplatnih alata za rudarenje podacima a neki od njih su RapidMiner, Weka, Orange, KNIME, AlphaMiner, R, MiningMart, KEEL, TANGARA, OpenNN i tako dalje. U nastavku su ukratko objašnjeni neki od ovi alata.

5.1.1. RapidMiner

RapidMiner prvobitno je poznat kao YALE (Yet Another Learning Environment), te je razvijen 2001. godine na tehničkom sveučilištu u Dortmundu. Tek 2007. godine preimenovan je u RapidMiner. To je softverska platforma koja je razvijena od tvrtke koja se zove isto kao i sam alata, RapidMiner te je pisana u Javi. Ovaj alat pruža integriranu okolinu za strojno učenje, prediktivne analize, rudarenje teksta, te rudarenje podataka. Ovaj alat podržava sve procese rudarenja podataka kao što su pripremanje podataka, vizualizacija odnosno prikaz rezultata, validacija i optimizacija. Postoji osnovna verzija koja je besplatna te tzv. Professional Edition koja je komercijalna.

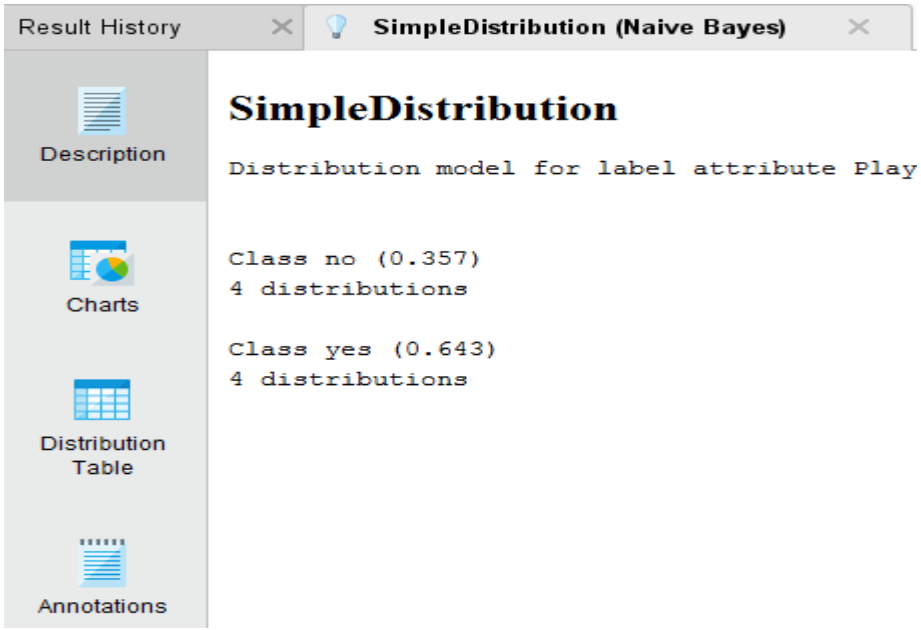


Slika 15. korisničko sučelje programa RapidMiner

Na slici je prikazano korisničko sučelje programa RapidMiner. U gornjem lijevom dijelu sučelja kod naziva „Repository“ se nalaze naši repozitoriji. U tom dijelu programa možemo kreirati novi repozitorij u koji ćemo smjestiti dvije mape, jednu za procese a drugu za podatke. U mapu za podatke dodajemo podatke koji smo spremili na našem računalu, pritiskom na „Add Data“. U mapu procesa spremamo procese koji se dodaju u pogled dizajna iz dijela gdje su naznačeni procesi. U pogledu „Design“ dodajemo proces i filtere te ostale radnje kako bismo obradili određeni skup podataka, a u pogledu „Result“ vidimo rezultate obrade podataka.

RapidMiner ima ugrađen skup primjera, odnosno supove podataka. Skup podataka nad kojim je provedena analiza je Golf. Metoda koja je korištena jest Naivni Bayesov klasifikator. Ovaj skup podataka sadrži podatke o vremenu na osnovu kojih se može odrediti jesu li vremenski uvjeti pogodni za igranje golfa. Skup podataka se sastoji od 5 atributa, a to su Outlook, Temperature, Humidity, Windy i Play, te 14 instanci. Proces započinjemo dodavanjem skupa podataka na radnu podlogu RapidMiner-a, odnosno sa Retrieve operatorom koji učitava odabrani skup podataka

iz repozitorija. Ovaj skup podataka sadrži i kontinuirane i diskretne varijable te iz tog razloga povezujemo Numerical to Polynominal operator na Retrieve operator. Taj operator pretvara kontinuirane varijable u diskretne, u ovom slučaju to su atributi Humidity i Temperature. Sljedeći korak je povezivanje na *Set Role* operator kako bismo identificirali, odnosno odabrali klasni atribut u ovom slučaju je to atribut *Play.* Sljedeći korak jest dodavanje *Valdiation* operatora kako bismo izgradili model i provjerili točnost modela. Naive Bayes operator smještamo u prostor za treniranje, a *Apply Mode*l i *Performance* operator u prostor za testiranje. *Naive Bayes* operator generira Naivni Bayesov klasifikacijski model. Taj klasifikator je baziran na primjeni Bayesovog teorema uvjetne vjerojatnosti sa pretpostavkom nezavisnosti. To znači da Naivni Bayesov klasifikator pretpostavlja da prisustvo ili odsustvo određenog obilježja klase nije povezano sa prisustvom ili odsustvom bilo kojeg drugog obilježja. *Apply Model* operator primjenjuje već naučeni ili trenirani model za stvaranje predviđanja, dok se Performance operator koristi za izvođenje statističke procjene zadatka klasifikacije. Izlaz iz ovog operatora dostavlja *Performance Vector,* odnosno listu izvedenih vrijednosti kriterija.



Slika 16. Prikaz rezultata u RapidMiner-u.

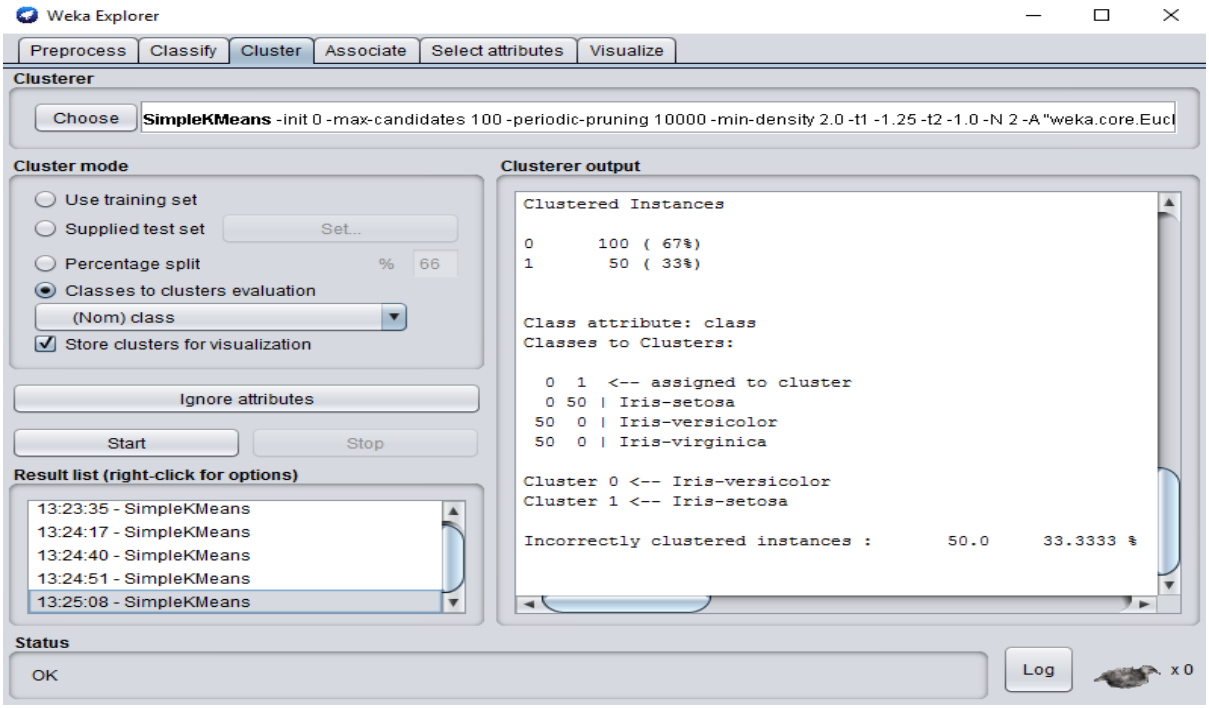
Nakon izvršene analize u prostoru za rezultate možemo vidjeti rezultate analize kao što je prikazano na slici. Podaci iznose 0.357 za klasu *no* i 0.643 za klasu *yes.* Na osnovu rezultata analize možemo zaključiti da s obzirom na vremenske uvijete vjerojatnost da je pogodno vrijeme za igrati golf iznosi 64.3% , a da nije 35.7%.

5.1.2. Weka

Weka (Waikato Environment for Knowledge Analysis) je poznat softver napisan u Javi, razvijen je na sveučilištu Waikato na Novom Zelandu. Ovaj alat sadrži kolekciju vizualizacijskih alata i algoritama za analizu podataka i prediktivno modeliranje. Weka podržava nekoliko standardnih zadataka rudarenja podataka, kao što su pred obrada, klasteriranje, klasifikacija, regresija, vizualizacija te selekcija obilježja. Prednosti ovog alata su slobodno korištenje pod GNU (General Public Licence) licencom, prenosivost iz razloga što je napisan u Javi te ga je moguće pokrenuti na gotovo svakoj modernoj platformi, opsežan skup pred obrade podataka te tehnika modeliranja te jednostavnost korištenja s obzirom na grafičko korisničko sučelje.

Metoda koja je korištena za analizu u programu Weka jest klasteriranje. Skup podataka koji je korišten prilikom ove analize je *Iris.arf. O*vaj skup podataka se sastoji od 5 atributa, a to su *sepallength, sepalwidth, petallength, petalwidth i class* atributi. Klase su *Iris-setosa, Iris-versicolor i Iris-virginica*. Broj instanci iznosi 150 od čega je 50 Iris-setosa, 50 Iris-versicolor te 50 Iris-virginica. Prvi korak jest učitavanje skupa podataka. Nakon učitavanja možemo vidjeti osnovne podatke o skupu podataka koji je učitan. Izabiremo *Cluster* kao metodu koju ćemo koristiti. Algoritam koji je korišten prilikom analize jest *SimpleKMeans.* Za prvu analizu *k* vrijednost je postavljena na 2 što znači da željeni broj klastera iznosi 2. Način klasteriranja je postavljen na *Use training set* što znači da nakon generiranja klasteriranja instance se svrstavaju u klastere u skladu s odgovarajućim klasterom te se računa postotak instanci koje pripadaju svakom klasteru. Nakon završenog procesa klasteriranja u *Cluster output* prozoru možemo vidjeti rezultate analize. Slučaj kada je broj klastera postavljen na 2, u prvom klasteru je grupirano 100 instanci što je 67%, dok je u drugom klasteru 50 instanci odnosno 33%. Broj iteracija iznosi 7, a kvadratna pogreška iznosi 62. 14. S

obzirom na veliku kvadratnu pogrešku napravljena je još jedna analiza gdje je *k* postavljen na 3. Slučaj kada je broj klastera postavljen na 3, u prvom klasteru grupirano 50 instanci, u drugi klaster 50 instanci te u treći klaster 50 instanci što daje postotak od 33% po klasteru. Broj iteracija iznosi 3, a kvadratna pogreška za ovaj slučaj iznosi 7.82.



Slika 17. Prikaz sučelja programa Weka

Na slici 17 prikazano je sučelje za klasteriranje u programu Weka. Sa lijeve strane je prikazan izbornik za odabir algoritma, odabir načina klasteriranja te gumb Start za početak klasteriranja. Sa desne strane možemo vidjeti izlaz klasteriranja odnosno rezultat izkojeg može pročitati podatke.

Sljedeća analiza biti će izvedena primjenom istog SimpleKMeans algoritma te na istom skupu podataka. Razlika će biti u načinu klasteriranja koji će u ovom slučaju biti postavljen na Classes to cluster evaluation. Kod primjene ovog načina prvo se ignorira klasni atribut te se generira klasteriranje. Za vrijeme faze testiranja dodjeljuju se klase klasterima, na osnovu većinske vrijednosti klasnog atributa unutar svakog klastera. Za prvu analizu k iznosi 2. U rezultatima možemo vidjeti da kvadratna pogreška iznosi

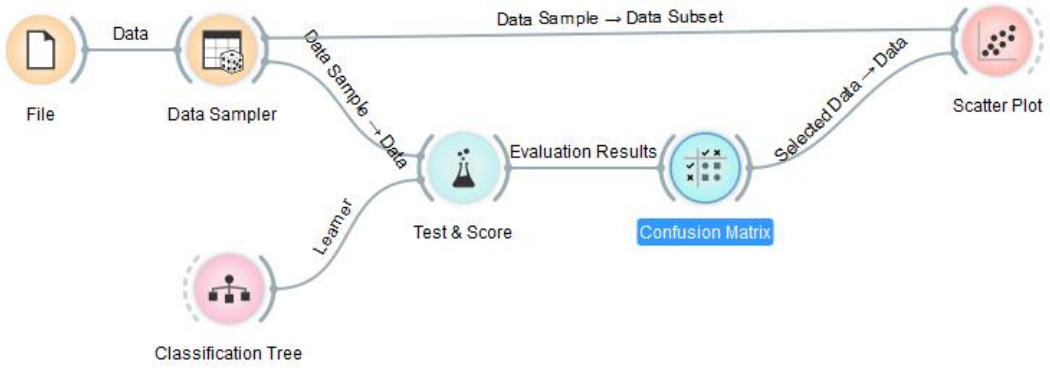
12.14, a broj iteracija iznosi 7. u konfuzijskoj matrici možemo vidjeti da prvi klaster kojem je u ovom slučaju dodijeljena klasa Iris-versicolor sadrži 50 instanci klase Iris-versicolor te 50 instanci klase Iris-virginica što je ukupno 100 instanci dodijeljenih u prvi klaster, odnosno 67%. Drugom klasteru dodijeljeno je 50 instanci klase Iris-setosa, što je 33%. možemo vidjeti da broj netočno klasteriranih instanci iznosi 50, odnosno 33%. Za sljedeću analizu k iznosi 3, način klasteriranja je Classes to cluster evaluation. U ovom slučaju kvadratna pogreška iznosi 6.99, a broj iteracija je 6. Iz rezultata možemo vidjeti da je u prvi klaster kojemu je dodijeljen Iris-versicolor klasterirano 47 instanci vrste Iris-versicolor te 14 instanci vrste Iris-virginica, u drugi klaster kojemu je dodijeljen Iris-setosa, klasterirano je 50 instanci vrste Iris-setosa, te u treći klaster kojemu je dodijeljen Iris-virginica klasterirano je 36 instanci vrste Iris-virginica te 3 instance vrste Iris-versicolor. Konačno možemo vidjeti da broj netočno klasteriranih instanci iznosi 17, odnosno 11.33%.

5.1.3. Orange

Orange je besplatni softver za strojno učenje i rudarenje podataka napisan u Python-u. program je razvijen i sačuvan u bio informatičkom laboratoriju na fakultetu za računalstvo i informatiku u Ljubljani. Softver je baziran na komponentama vizualnog programiranja. Komponente se zovu „widgeti“ te imaju raspon od jednostavne vizualizacije podataka, odabira pod skupova i pred obrade, do empirijske evaluacije učenja algoritama i prediktivnog modeliranja. Vizualno programiranje je implementirano kroz sučelje u kojem su radni sljedovi (eng. workflows) kreirani povezivanjem predefiniranih „widgeta“ ili widgetima koje su dizajnirali korisnici. Napredni korisnici mogu koristiti ovaj softver kao Python zbirku (eng. library) za manipuliranje podacima te izmjenu widgeta.

Metoda koja je korištena u programu Orange je metoda stabla odlučivanja. Skup podataka koji je korišten je Iris.tab. Skup podataka se sastoji od četiri obilježja, a to su sepal length, sepal width, petal lenght i petal width, te diskretne klase sa tri vrijednosti, Iris-setosa, Iris-versicolor te Iris-virginica. Ukupan broj instanci u ovom skupu podataka iznosi 150. Kao što je prije objašnjeno Orange koristi widgete za realizaciju radnog

slijeda rudarenja podataka. Proces započinje File widget-om u koji učitavamo odabrani skup podataka. Sljedeći korak je povezivanje File widget-a sa Data Sampler widgetom. Taj widget služi za postavljanje vrijednosti uzorka, odnosno u njemu možemo odabrati podskup instanci iz ulaznog skupa podataka za klasifikaciju. U ovom primjeru 70% od ukupnog uzorka, što je za ovaj slučaj 105 instanci od ukupnih 150. Nakon što smo postavili podatke za treniranje, povezujemo Data Sampler i Classification Tree widget-e sa Test & Score. Taj widget nam služi za pregled rezultata testiranja, odnosno u njemu možemo pročitati točnost klasifikacije koja u ovom slučaju iznosi 93.3%. Na Test & Score povezujemo Confusion Matrix widget, u kojem se vrednuje rezultat, odnosno možemo vidjeti rezultate testiranja algoritma u obliku konfuzijske matrice. U ovom slučaju, za klasu Iris setosa 36 je instanci točno klasificirano od mogućih 36, za klasu Iris versicolor 34 od mogućih 38, a za klasu Iris virginica 28 od 31. Klasifikacija je provedena na uzorku od 105 instanci.



Slika 18. Prikaz radnog slijeda u programu Orange

Na slici možemo vidjeti prikaz radnog slijeda prilikom korištenja stabla odlučivanja, te widgete koji su navedeni i objašnjeni prilikom objašnjenja gradnje stabla odlučivanja u Orange-u. Scatter Plot widget služi za vizualizaciju atributa. Podaci su prikazani u obliku skupa točaka, od kojih svaka ima vrijednost na x i y osi.

6. ZAKLJUČAK

Rudarenje podataka se primjenjuje kako bismo iz velike količine podataka došli do zanimljivih i potrebnih informacija ili pak znanja. Kako bi rudarenje podataka bilo moguće postoji čitav niz metoda za rudarenje podacima. S obzirom na široku primjenu tog područja neke metode nisu deklarirane kao isključive metode rudarenja podataka, no u literaturi se naravno spominju i one koje spadaju u područje rudarenja podataka. U ovom radu navedene su neke od metoda. Stabla odlučivanja te Naivni Bayesov klasifikator spadaju među najpoznatije te najkorištenije metode za rješavanje problema klasifikacije i predikcije. Ukoliko postoji potreba za razvrstavanjem, odnosno grupiranjem uzoraka u predefinirane grupe tada koristimo metode za klasteriranje kao što su hijerarhijsko klasteriranje ili K-means klasteriranje. U ovom radu spomenuto je i memorijski temeljno razlučivanje koje je građevni element algoritma K-means metode klasteriranja, te metoda potrošačke košarice koja spada pod asocijacijska pravila te se najčešće koristi za otkrivanje povezanosti unutar transakcija koja se tiču prodaje robe u maloprodajnim centrima.

Alati za rudarenje podataka korišteni za prikaz metoda rudarenja podataka u ovom radu su RapidMiner, koji je jedan od najkorištenijih alata za rudarenje podataka, Weka te Orange. U RapidMineru je korišten Naivni Bayesov klasifikator kod kojeg su atributi unutar podataka međusobno neovisni. Cilj rudarenja je bio na osnovu Bayesovog teorema uvjetne vjerojatnosti predvidjeti da li će se igrati golf s obzirom na vremenske uvijete. U Orange-u je prikazana metoda stabla odlučivanja te set podataka koji je korišten je Iris.tab. U Weki je korištena metoda klasteriranja te algoritam SimpleKMeans. Provedene su 4 analize na dva različita načina klasteriranja sa po 2 i 3 određena klastera.

Korištenjem ova tri alata može se vidjeti kako je vrijeme potrebno za rudarenje na ovakvim skupovima podataka vrlo malo. U slučaju kada skup podataka sadrži mnogo više instanci vrijeme se povećava, no i dalje štedi vrijeme koje bi bilo potrebno analitičaru da ručno obradi podatke. Samo korištenje alata za rudarenje ne oduzima toliko vremena koliko i priprema podataka.. Stoga je vrlo bitno odvojiti vrijeme za kvalitetnu pripremu podataka kako bi i sama primjena alata i metoda za rudarenje bila jednostavnija, kvalitetnija i brža.

LITERATURA

Knjige:

1. Kantardžić, M., *Data Mining concepts, models, methods, and algorithms.* John WIley & Sons, Inc., Hoboken, New Jersey, 2011.
2. Klepac, G. i L. Mršić, *Poslovna inteligencija kroz poslovne slučajeve,* Biblioteka Poslovna znanja, Lider press.
3. Panian, Ž. i G. Klepac , *Poslovna inteligencija*, Masmedia, Zagreb, 2003.
4. Han, J. i M. Kamber, *Data mining: Concepts and Techniques Second edition*, Morgan Kaufmnn, Elsevier, 2006.

Internet izvori:

1. Klepac, Goran – osobne stranice, „Što je to data mining?“, http://www.goranklepac.com/index.asp?j=HR&iz=1&sa=1&vi=1&hi=1

2. What is R: Intorduction to R, The R environment, https://www.r-project.org/about.html, (Pristupljeno: 14.06.16, 15:12)

3. TechTarget: Examining the KNIME open source data analytics platform, http://searchbusinessanalytics.techtarget.com/feature/Examining-the-KNIME-open-source-data-analytics-platform (Pristupljeno: 14.06.2016, 15:58)

4. Knowledge extraction based on evolutionary learning, http://www.keel.es/, (Pristupljeno: 16.06.2016, 16:10)

5. KDnuggets: R leads RapidMiner, Phyton atches up, Big Data tools grow, Spark ignites, http://www.kdnuggets.com/2015/05/poll-r-rapidminer-python-big-data-spark.html, (13.06.2016, 11:38)

6. THENEWSTACK, Six of the Best Open Source Data Mining Tools, http://thenewstack.io/six-of-the-best-open-source-data-mining-tools/

7. KDnuggets, Software Suites/Platforms for Analytics, Data Mining, & Data Science, http://www.kdnuggets.com/software/suites.html

Popis slika

Popis tablica